Journal of Guidance, Control, and Dynamics, Volume 46, Issue 8, August 2023, pp. 1564-1579
DOI: 10.2514/1.G006804

A Collision-Avoiding Model Predictive Rendezvous Strategy to Tumbling Launcher Stages

Jesús Ramírez* and Leonard Felicetti†
Cranfield University, Cranfield, United Kingdom, MK43 0AL

Damiano Varagnolo‡
Norwegian University of Technology and Science in Trondheim, Trondheim, Norway, NO-7491
and University of Padova, Italy

The paper considers the situation where a small satellite shall autonomously rendezvous with a tumbling object in a circular Low Earth Orbit (LEO) and derives a path-based Model Predictive Controller that uses the docking point state and position of the chaser to guide it to a safe docking autonomously. The strategy embeds collision avoidance elements and reduces the computational effort for calculating the pulses to be provided by the thrusters through opportune algebraic manipulations, a Runge-Kutta 4 propagation method using linearized state transition matrices, and implicit embedding of dynamically equivalent thrust models, leading to constant state propagation matrices. Furthermore, the inputs design optimization problem and the embedded collision avoidance scheme are modeled and explicitly crafted as convex problems, contributing positively to low computational requirements. The docking and collision avoidance capabilities of the proposed scheme are extensively tested in an environment that accounts for all the perturbations relevant to LEO frameworks, for realistic thrust schemes, and for uncertainties in the measurement. Numerical results assess which tumbling objects can be docked or not by means of the proposed schemes as a function of the tumbling rates vs. the thrust capabilities & hardware uncertainty of the docker.

*Cranfield student, School of Aerospace, Transport, and Manufacturing, jesusframirezsanchez98@gmail.com

†Corresponding author. Lecturer in Space Engineering at Cranfield University, School of Aerospace, Transport and, Manufacturing, Cranfield University, Cranfield, Bedfordshire MK43 0AL, Leonard.Felicetti@cranfield.ac.uk

‡Professor in Big Data Cybernetics, Department of Engineering Cybernetics, NTNU - Norwegian University of Technology and Science in Trondheim, Trondheim, Norway, and Department of Information Engineering, University of Padova, Italy, damiano.varagnolo@ntnu.no

Published by American Institute of Aeronautics and Astronautics. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial License (CC:BY:NC 4.0).
The final published version (version of record) is available online at DOI:10.2514/1.G006804. Please refer to any applicable publisher terms of use.

Nomenclature

Acronyms		
CSM	=	Command and Service Module
DCM	=	Direction Cosine Matrix
DoF	=	Degrees-of-Freedom
HCW	=	Hill-Clohessy-Wiltshire
LEO	=	Low Earth Orbit
LM	=	Landing Module
LVLH	=	Local Vertical Local Horizon reference frame
MPC	=	Model Predictive Control
NTSM	=	Non-singular Terminal Sliding Mode
STM	=	State Transition Matrix

Variables		
$\overline{\Delta}$	=	Pulse thrust maximum duration [s]
Δ	=	Pulse thrust duration [s]
$U_{\max}$	=	Thrust limit [N]
n	=	Mean motion [rad/s]
μ_E	=	Earth gravitational parameter [m^3s^{-2}]
m_{chaser}	=	Chaser mass [kg]
$\theta_{\text{debris length}}$	=	Debris half length [m]
$\theta_{\text{debris radius}}$	=	Debris radius [m]
L_c	=	Distance from border to center of debris [m]
$\boldsymbol{r}$	=	Position vector [m]
d	=	Relative distance [m]
$\boldsymbol{v}$	=	Velocity vector [m/s]
v	=	Relative velocity [m/s]
$\boldsymbol{\omega}$	=	Angular velocity vector [rad/s]
ω_{yz0}	=	Rotating rate [deg/s]
I_x	=	Axial body inertia [kgm^2]
I_y	=	Transverse body inertia [kgm^2]
$\boldsymbol{x}$	=	State vector of a time step [m; m/s]
$\boldsymbol{X}$	=	Vector containing the state vectors of a horizon [m; m/s]

$\boldsymbol{u}_v$	=	Control vector of a time step [N]
$\boldsymbol{U}_v$	=	Vector containing the control actuation over the horizon [N]
α_ψ	=	Turning angle for generating collision avoidance path [rad]
ε_p	=	Tolerance in docking distance [cm]
ε_v	=	Tolerance in docking velocity [cm/s]
I	=	Total Impulse [Ns]
μ_D	=	Mean value of docking distance results [m]
μ_V	=	Mean value of docking velocity results [cm/s]
μ_I	=	Mean value of total impulse results [Ns]
σ_D	=	Standard deviation of docking distance results [m]
σ_V	=	Standard deviation of docking velocity results [cm/s]
σ_I	=	Standard deviation of total impulse results [Ns]
Index		
j	=	Time index for a specific slot in horizon
k	=	Time index in discrete-time settings
t	=	Time index in continuous-time settings
Control parameters		
α	=	Sizer of approach sigmoid slope
γ	=	Sizer of approach sigmoid half point
ψ	=	Initial angle driver for synchronization phase
$\theta_{\text{safety factor}}$	=	Sizer of safety sphere
$\theta_{\text{end phase}}$	=	Sizer of distance for end phase
α'	=	Sizer of end sigmoid slope
$\boldsymbol{Q}$	=	Weight matrix for state
$\boldsymbol{R}$	=	Weight matrix for control
N_{step}	=	Number of steps in horizon
ΔT	=	Time step [s]
Subscripts		
x	=	x component of a vector
y	=	y component of a vector
z	=	z component of a vector
0	=	Initial condition

chaser	=	Chaser
target	=	Target
dock	=	Docking point
v	=	Virtual
reference	=	Reference path
appr	=	*Approach* reference path
syn	=	*Synchronization* reference path
end	=	*End* reference path
Superscripts		
I	=	Referred to the inertial reference frame
L	=	Referred to the LVLH reference frame

I. Introduction

The problem of rendezvous with non-cooperative tumbling targets is one of the main milestones to overcome if we are to be successful in active debris removal. The scenario poses a challenge for the design of Guidance Navigation and Control (GNC) algorithms. The problem gets much harder to deal with when low computation capabilities and control actuation limitations are put into play. This article focuses on the scenario where a small satellite of such limited capabilities, e.g., a CubeSat, performs a rendezvous towards a large piece of debris randomly rotating in the perturbed Low Earth Orbit (LEO) environment.

Debris in space is affected by external torques such as atmospheric drag or solar pressure. Indeed, without any kind of control or spin stabilization, they are likely to be randomly tumbling at the time of the rendezvous [1]. CubeSat-like chasers that are supposed to dock at these objects autonomously shall moreover implement control solutions that are able to handle uncertainties, perturbations, and high tumbling rates at a very low computational and actuation cost. In addition, to avoid risks contributing to Kessler syndrome, such control strategies shall guarantee collision avoidance.

This paper proposes a control algorithm that uses Model Predictive Control (MPC) and Convex Optimization to address these challenges. The method includes the generation of sigmoid-shaped paths towards the target, embedding collision avoidance strategies using information about the current state of the docking point. This approach not only improves the performance from a control point of view but also has good computational properties, as it reduces the calculation effort thanks to a simpler formulation without complex constraints and using linearization and convexification techniques.

As for the target selection, we note that rocket bodies constitute about 11% of cataloged objects [2]. Furthermore, such pieces of debris are considered dangerous due to the size and the energy eventually stored inside them and require

a lot of time to decay and re-enter the atmosphere naturally, given their low area-to-mass ratio. For these reasons, we selected the removal of rocket bodies as the objective of the study in this paper. However, the strategies presented can be extended to other kinds of objects cluttering space.

The application of MPC to the rendezvous problem has been approached by several authors. For a general overview of possible applications and advantages of MPC, we refer to [3], which assesses MPC's potential to effectively address the challenges in the control of aerospace systems. The same reference provides a theoretical foundation of MPC, discusses the design aspects such as system modeling, constraint formulation, cost function definition, etc., and in particular analyses the problem of flight safety. We note that there exists a long list of articles dealing with cooperative proximity operations among spacecraft, i.e., situations where the propulsion systems on two spacecraft can cooperate towards completing the mission. An example of this is [4], which introduces strategies to improve the computational efficiency of MPC when applied to the problem of autonomous rendezvous and proximity maneuvering using low-thrust propulsion systems.

More related to our paper are those strategies that deal with the non-cooperative scenario of approach and docking. This is, for example, the case considered in [5], where the authors adopt MPC strategies for rendezvous with a tumbling stage. The autonomous onboard orbit control strategy is divided into four stages, each involving a sequence of low-thrust orbital transfer and rendezvous maneuvers. Interestingly, this framework is usable also for other purposes like following a predefined glide slope toward the target. The authors in [5] focus on a continuous low thrust scheme, while our paper deals with the case of pulse thrust technologies. We also notice that [5] promotes collision avoidance by linearizing the associated constraint, while we propose to compute dedicated reference paths that ensure collision avoidance.

On the other hand, the authors in [6] propose a Linear Quadratic MPC for obtaining optimal rendezvous trajectories where constraints and objectives are modified dynamically depending on the target distance. The article presents different test case scenarios considering docking corridors while assuming that the target is stabilized. In other words, the problem of uncooperative rendezvous to tumbling objects is not addressed in [6]. However, it does include aspects that are not evaluated in the here proposed paper, like the attitude control of the chaser or obstacle in the path towards the target.

Another example of fast optimization obtained through convexification is given by the work in [7], where a successive convexification method is presented to solve the cooperative rendezvous problems between Landing Module (LM) and Command and Service Module (CSM) of the Apollo mission. Such a paper considered a full 6 Degrees-of-Freedom (DoF) model for both the spacecraft, as well as propulsion system providing a finite-duration impulsive thrust, plume impingement constraints, and fixed approach corridors to specific target points of the target, all included as constraints in the optimization problem definition. Even if [7] develops and solves the optimization problem using an open-loop strategy and not using an MPC approach, we note that some key factors need to be considered in our formulation, such as the inclusion of finite-duration thrust bits and strategies that could avoid collision with the target. Our approach

indeed includes these two features in the model used for optimizing the 3 DoF trajectory of the chaser. However, we do not apply them to a full 6 DoF model because of the different time scales of the relative orbital and attitude motion problems. This difference hinders the full state MPC application to such long horizons, especially due to the tumbling motion which cannot be predicted with such anticipation.

The study presented in [8] also relates to our proposal. Here, MPC is used for transfer guidance to the ISS using an approach that divides, as we propose, the problem into separate translational and associated attitude trajectory design phases that operate sequentially. However, [8] models the problem without taking into account collision avoidance requirements: the objective is just to pursue a final attitude and position. Thus, the authors there tune the controller so as to maintain the chaser in the right docking corridor. The authors in [9] propose a MPC strategy for spacecraft control, dedicated though in this case to rendezvous hovering. The purpose of the controller in [9] is thus to reach and remain on a periodic trajectory inside a given box-type hover zone while minimizing fuel consumption. A mission that differs from the one considered here. Nevertheless, the proposed approach here is technically interesting and related to our strategy since it accounts for actuation saturation constraints. The research in [10] instead focuses exactly on the same mission we do: docking with a tumbling target. The paper presents a variable-horizon guidance scheme whose formulation enables the identification of docking opportunities. We note that this is a logical step that is also present in our paper, even if we propose to compute favorable docking instants in a different way. Similarly to us, [10] also formulates the model predictive control using convex formulations of the cost functions. The structure of the rendezvous maneuver and the structure of the collision avoidance are though quite different, ours containing a more sophisticated -yet effective- sequence whose purpose is to incorporate the collision avoidance constraint in the planning of the mission directly.

Computational and actuation capabilities challenges presented in our paper are also addressed in [11]. The authors propose using an MPC algorithm to rendezvous a chaser that is very limited in its thrust possibilities, and highlight the effects of choosing different prediction horizon lengths in a situation where the chaser is within a perturbed environment. However, the MPC formulation in [11] does not embed collision avoidance considerations.

Related to this, we note that the intrinsic capacity of handling predictions of MPC approaches is beneficial and leads to improvements against other controllers, such as Linear Quadratic Regulators when the model for propagating the predictions is accurate enough for control purposes. Typically, the more the model accounts for effects and disturbances actuating on the system, the better MPC performs, even if generally at the cost of increased complexity in tuning the models and in computational efforts for finding the actuation signals.

Some MPC schemes were tested onboard the International Space Station [12–14] or in dedicated test-bed facilities on ground [15, 16]. These tests also considered tumbling targets [12, 15, 16], chasers under under-actuated conditions due to failures [13], and uncertain path constraints [14]. In all these implementations, the key driver was represented by the computation capabilities of the test devices, which often limited the potentialities of more sophisticated algorithms [15, 16].

We then note that for limited satellites such as CubeSats, MPC approaches need to embed the simplest prediction propagation schemes that still guarantee successful autonomous docking. Towards this goal, in this work, we propose to use *Hill-Clohessy-Wiltshire* (HCW) equations in a way that leads to a convex optimization problem and a formulation that may also address collision avoidance. The proposed strategy is comparable with the one in [17], where the authors state that *MPC cannot directly enforce a nonlinear spherical keep-out zone for collision avoidance but instead use a separating hyperplane that lies on the sphere boundary and enables the solution set to remain convex* [18]. In this paper, we prove that keeping out of this sphere is possible, although with some limitations. Thus, we make use of concepts similar to the ones in [17] but arriving at a different MPC formulation, with the same conclusion as in [17] that convexification of the control problem is a viable strategy for achieving autonomous docking with a reduced computation effort. Moreover, with respect to [17], we test a larger variety of tumbling rates of the target, of perturbations, of noises, plus simulate a more realistic pulse technology than the continuous one used in [17]. From a results perspective, our scheme achieves similar levels of ΔV required for similar levels of maximum control.

The problem of guaranteeing collision avoidance with the tumbling target is thus not new. For example, [19] proposes a control algorithm that joins a *non-singular terminal sliding mode* (NTSM) together with a collision avoidance strategy that makes use of motion planning and a *danger zone*. Thus, an approach presenting some similarities with the strategy we propose below. The authors in [19] express that *the magnitude of the needed control inputs are less than that in the existing literature.* Nevertheless, this control is not constrained in the analyzed scenario, and then the maximum obtained set as a requirement. In our paper, we instead develop a control strategy that guarantees docking and collision avoidance in more demanding scenarios even if investing a smaller control effort, i.e., maximum thrust available.

Thus, summarizing, with respect to the existing literature our main contributions stand on introducing a novel MPC formulation for docking with a tumbling stage that embeds:

1) analytical expressions to reduce the computational effort in the prediction step,
2) models of the dynamics of the satellites that limit considerably the predictions propagation error,
3) impulsive thrust profiles with improved fidelity, and
4) a collision avoidance strategy that guarantees the convexity of the control problem by means of computing and refining opportune sigmoid-shaped safe approaching paths towards the tumbling object.

The proposed approach does not only improve the performance from the point of view of the control but is shown to also have good computational properties, in the sense of reducing the computation effort thanks to the avoidance of complex constraints, linearization and convexification of the control problem. We furthermore demonstrate by simulation that with the designed GNC strategy a CubeSat of 20 kg with 500 mN of thrust in each axis is able to dock with a spent Kosmos (SL-8) upper stage tumbling at 3 deg/s in a typical LEO environment, which constitutes a more demanding scenario for a more restricted chaser than the one evaluated in [19].

The rest of the document is structured as follows: Section II describes the rendezvous problem and summarizes the

physical equations of motion considered throughout the paper. Sections III and IV propose our MPC formulation and the collision avoidance strategy, respectively. Section V describes the simulation setup, followed by Section VI which presents a series of results from different simulations. Section VII concludes with a summary of the learned points.

II. Equations of motion

We consider a CubeSat-like chaser that shall dock a tumbling spent upper-stage target in a circular Low Earth Orbit, as represented in Figure 1. Without loss of generality for the sake of this paper, we assume the target to be a cylinder with uniformly distributed mass and center of mass coincident with the geometric center. The cylinder is $2\theta_{\text{debris length}}$ long and with a radius $\theta_{\text{debris radius}}$, as indicated in Figure 1. The objective of the mission is to dock with the target at a specific desired point and at the lowest relative velocity possible. Again without loss of generality, we assume this docking point to be located in the furthest extreme from the center of mass of the target, as highlighted by the green circle in Figure 1.

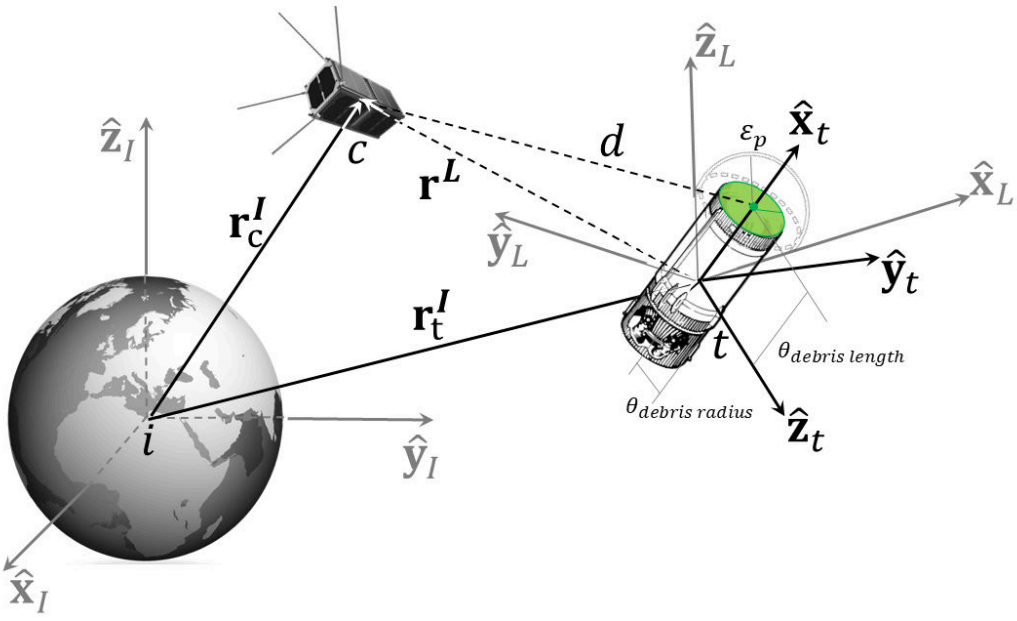


Fig. 1 Representation of the mission scenario concept.

The target center position referred to an inertial Earth frame with axes $\hat{x}_I$, $\hat{y}_I$, and $\hat{z}_I$ is indicated with the vector $\boldsymbol{r}^I_{\text{target}}$. For the scope of this study, this orbit is assumed to be circular, with a mean motion defined as

$$n := \sqrt{\frac{\mu_E}{\left|\boldsymbol{r}^I_{\text{target}}\right|^3}} \tag{1}$$

with $|\cdot|$ the Euclidean norm of the vector and μ_E the gravitational parameter of the Earth. The circular orbit approximation is in any case valid for almost all the target objects in LEO [2].

The reference frame used for the rendezvous analysis and control design is thus located in the center of mass of the target body so that it follows the same orbit of the object to rendezvous with. This Local Vertical Local Horizon (LVLH) frame has thus its axes $\hat{x}_L$, $\hat{z}_L$ and $\hat{y}_L$ oriented respectively as along the radial direction, as the orbit normal, and as in the same plane of the orbit of the target, perpendicular to the other two directions and therefore pointing as the velocity of the target. This means that we can define the angular momentum vector of the reference orbit of the target expressed in the LVLH frame as:

$$\boldsymbol{n}^L \coloneqq [0, 0, n]^T . \tag{2}$$

The chaser center position expressed in the inertial frame is indicated with the vector $\boldsymbol{r}^I_{\text{chaser}}$. In the practice, the control scheme developed in the remainder of the paper will consider affecting the relative position vector with respect to the LVLH frame, i.e.,

$$\boldsymbol{r}^L \coloneqq \boldsymbol{R}^L_I \left(\boldsymbol{r}^I_{\text{chaser}} - \boldsymbol{r}^I_{\text{target}} \right), \tag{3}$$

where $\boldsymbol{R}^L_I$ is the Direction Cosine Matrix (DCM) that transforms the vector components from the inertial reference frame $(\cdot^I)$ into the LVLH frame $(\cdot^L)$.

The control scheme makes use also of the relative velocity vector in the LVLH frame, i.e.,

$$\boldsymbol{v}^L \coloneqq \boldsymbol{R}^L_I \left(\boldsymbol{v}^I_{\text{chaser}} - \boldsymbol{v}^I_{\text{target}} \right) + \boldsymbol{n}^L \times \boldsymbol{R}^L_I \left(\boldsymbol{r}^I_{\text{chaser}} - \boldsymbol{r}^I_{\text{target}} \right). \tag{4}$$

This implies that the complete state of the system we intend to control is defined as

$$\boldsymbol{x}^L \coloneqq \begin{bmatrix} \boldsymbol{r}^L \\ \boldsymbol{v}^L \end{bmatrix}. \tag{5}$$

The chaser is modeled as a mass point; this means we ignore the attitude control problem. The target is modeled as a rigid body, hence having a full 6 DoF model. The orbital motion of the two objects is simulated by using the perturbed orbital equations expressed in LVLH frame [20]. More specifically, let m_{chaser} be the mass of the chaser, and assume it to be constant due to the low consumption of the maneuvers. Then $\boldsymbol{u}^L$ represents the forces applied by the actuators to the chaser, which will be modeled in detail in Section III.A. $\delta \boldsymbol{p}^L$, instead, refers to differential perturbations*. Finally,

*More precisely, the difference between the atmospheric drag force experienced by the target and that one experienced by the chaser, the same type of difference of the solar pressure, and the same for J2. As for estimating $\boldsymbol{\delta p}^L$ numerically, we send the reader forward to Section V.B, where we provide some implementation details.

the perturbed orbital equations are assumed to be

$$\delta \boldsymbol{a}^L \coloneqq \frac{d^2\boldsymbol{r}^L}{dt^2} = -2\boldsymbol{n}^L \times \boldsymbol{v}^L - \boldsymbol{n}^L \times \left(\boldsymbol{n}^L \times \boldsymbol{r}^L\right) - \frac{\mu_E}{\left|\boldsymbol{r}^L_{\text{target}} + \boldsymbol{r}^L\right|^3}\left(\boldsymbol{r}^L_{\text{target}} + \boldsymbol{r}^L\right) + \frac{\mu_E}{\left|\boldsymbol{r}^L_{\text{target}}\right|^3}\boldsymbol{r}^L_{\text{target}} + \frac{\boldsymbol{u}^L}{m_{\text{chaser}}} + \boldsymbol{\delta p}^L. \quad (6)$$

We assume that torques from environmental disturbances do not produce significant effects on the tumbling motion of the target for the duration of the docking maneuver. Therefore, we assume the target as following a free rotation motion characterized by the initial conditions to be treated as parameters in our simulations section, and thus so that:

$$\omega_{x0}, \quad \omega_{y0}, \quad \omega_{z0}, \quad \omega_{yz0} \coloneqq \sqrt{\omega_{y0}^2 + \omega_{z0}^2} \quad (7)$$

and an analytical solution for the target attitude motion as given in [21], i.e.,

$$\begin{cases} \omega_{\text{target},x}(t) & = \omega_{x0} \\ \omega_{\text{target},y}(t) & = \omega_{yz0} \sin\left(\omega_{x0}(1 - I_x/I_y)t + \tan^{-1}(\omega_{y0}/\omega_{z0})\right) \\ \omega_{\text{target},z}(t) & = \omega_{yz0} \cos\left(\omega_{x0}(1 - I_x/I_y)t + \tan^{-1}(\omega_{y0}/\omega_{z0})\right) \end{cases} \quad (8)$$

with I_x and I_y being the axial and transverse moment inertia of the target, respectively.

From (8) we can derive the target angular rate with respect to the LVLH reference frame as:

$$\boldsymbol{\omega}^L_{\text{target}} = \boldsymbol{R}^L_{\text{target}}\boldsymbol{\omega}_{\text{target}}(t) - \boldsymbol{n}^L \quad (9)$$

where $\boldsymbol{R}^L_{\text{target}}$ is the direction cosine matrix relating the LVLH reference frame with the target body frame. Equations from (8) to (9) may be used to simulate the rotating motion of the target, and thus the position of the docking point in time. More in detail, if we let the position of the docking point in the target's body reference frame be

$$\boldsymbol{r}^{\text{target}}_{\text{dock}} \coloneqq [\theta_{\text{debris length}}, 0, 0]^T \quad (10)$$

then the docking point position in the LVLH reference frame in time can be calculated as:

$$\boldsymbol{r}^L_{\text{dock}} \coloneqq \boldsymbol{R}^L_{\text{target}}\boldsymbol{r}^{\text{target}}_{\text{dock}}. \quad (11)$$

This implies that the docking point velocity in the LVLH reference frame can be calculated as:

$$\boldsymbol{v}^L_{\text{dock}} \coloneqq \boldsymbol{R}^L_{\text{target}}\left(\boldsymbol{\omega}_{\text{target}}(t) \times \boldsymbol{r}^{\text{target}}_{\text{dock}}\right). \quad (12)$$

For simplicity of notation, in the following, we denote the target-docking point relative distance and target-docking point relative velocity be respectively

$$d \coloneqq \left| \boldsymbol{r}^L - \boldsymbol{r}^L_{\text{dock}} \right|, \qquad v \coloneqq \left| \boldsymbol{v}^L - \boldsymbol{v}^L_{\text{dock}} \right|. \tag{13}$$

We consider the maneuver to be successful if the chaser reaches the docking point or the lateral surface of the target with a relative velocity v lower than a user-defined threshold ε_v. Otherwise, the maneuver is cataloged as an impact. Therefore, mathematically the successful-docking conditions are given by the following two mutually exclusive conditions: if the docking actually happens at the desired docking point, then

$$d < \varepsilon_p, \quad \text{if} \quad \boldsymbol{r}^{\text{target}}_{\text{chaser}}(1) > \theta_{\text{debris length}} \tag{14}$$

where $\boldsymbol{r}^{\text{target}}_{\text{chaser}}(1)$ indicates the first component of the vector. Otherwise, when the docking is produced in a lateral surface, then

$$\left| \boldsymbol{r}^L \times \boldsymbol{r}^L_{\text{dock}} \right| = \theta_{\text{debris radius}}, \quad \text{if} \quad \boldsymbol{r}^{\text{target}}_{\text{chaser}}(1) \leq \theta_{\text{debris length}}. \tag{15}$$

For the relative velocity:

$$v < \varepsilon_v. \tag{16}$$

Specific values of ε_p and ε_v are given in the simulation setup section.

III. Control Algorithm Formulation

The proposed path planning and control strategy is based on an ad-hoc Model Predictive Control scheme that embeds some adjustments to speed-up the computational time, and to include collision avoidance checks. The MPC returns control signals that minimize a cost function that is evaluated in a discrete-time given horizon of $N_{\text{step}} \cdot \Delta T$ steps. Here ΔT represents the sampling period of the scheme. At a given discrete time step $k \in \mathbb{N}$, the optimization problem selects that actuation signal that makes the forecasted trajectory of the system, together with the actuation itself, optimal with respect to the given cost function. Once this actuation signal is computed, only its first sample is then actually used, before reinitializing the search for the next optimal actuation signal. This process introduces robustness against eventual perturbing actions that were not predicted by the controller [22]. Figure 2 shows an example of such a discrete time MPC implementation. In the following subsections, we will present a suitable implementation of such algorithm to the problem defined in Section II.

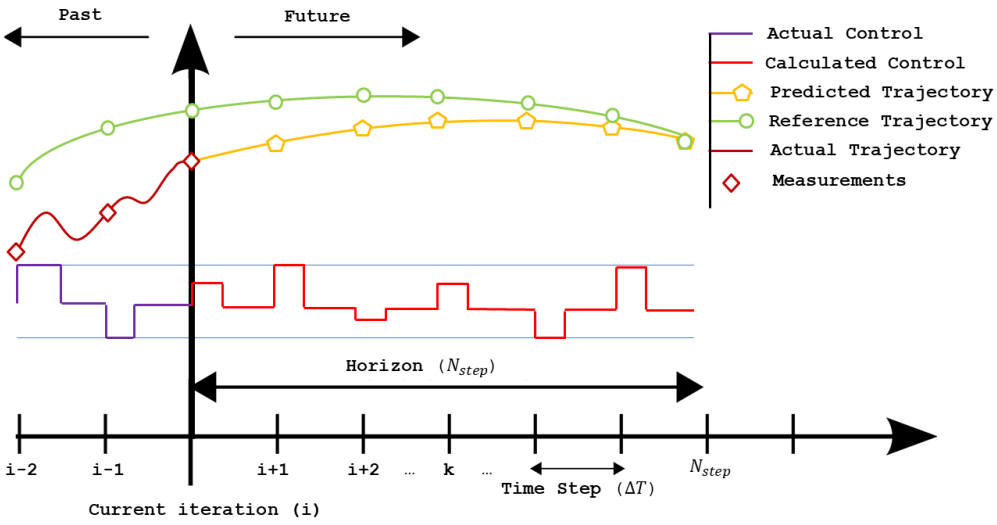


Fig. 2 MPC scheme.

A. Equivalent actuation model

We note that the current propulsion system technology for Cubesats is mostly based on cold-gas. This can provide impulse bits of different duration based on the actually commanded input [23], i.e., the actuation signals can be modeled as being a pulse-width modulation. Assuming that thrust can be provided in all the directions of the LVLH coordinate frame, we model such impulsive behavior of the actuation system as:

$$\boldsymbol{u}^L = \left[u_x^L,\ u_y^L,\ u_z^L\right]^T \tag{17}$$

subject to

$$\left|u_x^L\right| = U_{\max} \quad \text{or} \quad \left|u_x^L\right| = 0; \qquad \left|u_y^L\right| = U_{\max} \quad \text{or} \quad \left|u_y^L\right| = 0; \qquad \left|u_z^L\right| = U_{\max} \quad \text{or} \quad \left|u_z^L\right| = 0 \tag{18}$$

where $U_{\max}$ is the nominal thrust. At each discrete time step k we thus assume that the chaser applies a thrust of a fixed and constant nominal amplitude $U_{\max}$ for a time-step-dependent duration of

$$\boldsymbol{\Delta}(k) \coloneqq \left[\Delta_x(k),\ \Delta_y(k),\ \Delta_z(k)\ \right]^T \tag{19}$$

time units, as indicated schematically in Figure 3. Note that we implicitly assume every pulse time $\Delta(k)$ to be smaller than the sampling period ΔT. Moreover, given the specifics of the current cold-gas technology, we assume no errors-in-variables in the actuation, no delays, nor limitations on the minimum length of $\Delta(k)$.

We though note that the pulse-width modulation representation above is not directly prone to MPC implementations.

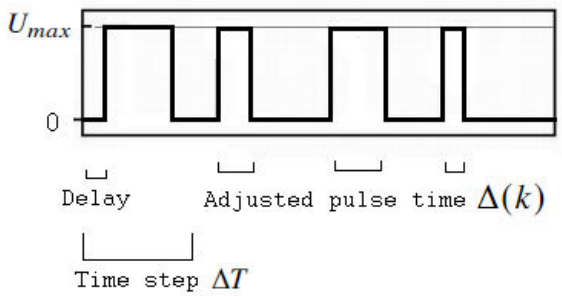


Fig. 3 Representation of a pulse-width modulation strategy of generating thrust on a specific LVLH direction.

Similar to [24], we thus transform such representation into a pulse amplitude modulation one as in Figure 4. In other words, we let the MPC scheme calculate a vectorial actuation signal for which the nominal thrust profile assumes the pulses to happen with a constant actuation frequency of f, constant duration $\overline{\Delta}$, but variable amplitude $\boldsymbol{u}_v^L = \left[u_{v,x}^L(k),\ u_{v,y}^L(k),\ u_{v,z}^L(k)\right]^T$ (still though limited by $U_{\max}$ in each component as shown in Figure 4).

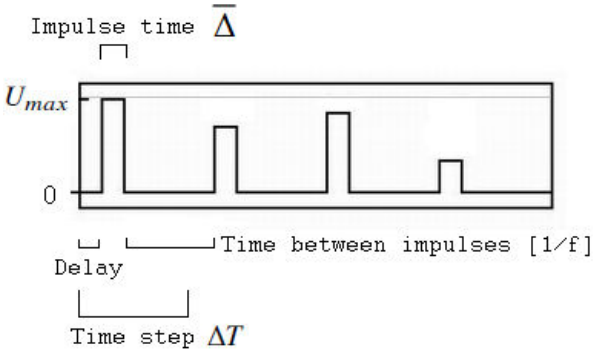


Fig. 4 Representation of a pulse-amplitude modulation strategy of generating thrust on a specific LVLH direction.

This means that after the MPC computes a reference actuation vector $\boldsymbol{u}_v^L(k)$, this needs to be translated to a spacecraft-ready actuation vector $\boldsymbol{u}^L(k)$ via the relations

$$\Delta_x(k) = \frac{\left|u_{v,x}^L(k)\right|}{U_{\max}}\overline{\Delta}, \qquad \Delta_y(k) = \frac{\left|u_{v,y}^L(k)\right|}{U_{\max}}\overline{\Delta}, \qquad \Delta_z(k) = \frac{\left|u_{v,z}^L(k)\right|}{U_{\max}}\overline{\Delta}. \tag{20}$$

Summarizing, the scheme ladders opportune translations between pulse-width-modulated thrust shapes, that are cold-gas propulsion systems compliant, and pulse-amplitude-modulated thrust shapes, that are MPC-scheme compliant. Such translations preserve the provided total impulse along the various directions of actuation.

We note though that computing $\boldsymbol{u}_v^L(k)$ and then translating it into $\boldsymbol{u}^L(k)$ is not equivalent to computing $\boldsymbol{u}^L(k)$ directly. However, we will show through opportune simulations that this approximation, given the type of missions considered in this paper, does not affect the performance of the maneuver. Furthermore, by selecting f and $\overline{\Delta}$ opportunely,

and by using the constraint in $U_{\max}$, the scheme ensures the actual control signal $\boldsymbol{u}^L(k)$ to never exceed its saturation limits.

B. Convexification of the constraints and cost function for the rendezvous problem

The proposed autonomous rendezvous scheme shall account for the dynamics of the bodies involved in the maneuver. The classical *Hill-Clohessy-Wiltshire* (HCW) equations [25] are used to propagate the state of the system over the prediction horizon. This enables considering linearized dynamics that have still powerful predictive capabilities while maintaining the convexity of the problem, so to keep the computational effort limited. Hence, the State Transition Matrix (STM) based on the original HCW equations, i.e.,

$$\Phi(t) = \begin{bmatrix} 4-3\cos(nt) & 0 & 0 & \frac{1}{n}\sin(nt) & \frac{2}{n}(1-\cos(nt)) & 0 \\ 6(\sin(nt)-nt) & 1 & 0 & \frac{2}{n}(\cos(nt)-1) & \frac{1}{n}(4\sin(nt)-3nt) & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \frac{1}{n}\sin(nt) \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin nt & 0 \\ 6n(\cos(nt)-1) & 0 & 0 & -2\sin(nt) & -3+4\cos(nt) & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix}, \tag{21}$$

enables propagating the trajectory of the chaser in the LVLH frame from a generic initial state $\boldsymbol{x}^L(t_0)$ (i.e., as in (5)) as

$$\boldsymbol{x}^L(t) = \Phi(t-t_0)\boldsymbol{x}^L(t_0) + \int_{t_0}^{t} \frac{1}{m_{\text{chaser}}}\Phi(\tau-t_0)\begin{bmatrix}\mathbf{0}\\ \boldsymbol{I}\end{bmatrix}\boldsymbol{u}_v^L(\tau)d\tau. \tag{22}$$

Here $\mathbf{0}$ is a 3×3 matrix of zeros, while $\boldsymbol{I}$ is the 3×3 identity matrix.

Inspired by [26], we then propose the following efficient approach to reduce the computational effort on predicting the effects of control actions on the dynamics: build a Runge-Kutta 4-like expansion for propagating the effects of the applied control on the system over the prediction horizon, and avoid iterative processes at each prediction step of the MPC thanks to a non-state dependent linearization.

By considering (20) and by modeling the force applied during the actuation time as constant for the duration of the impulse, the second term in (22), if integrated between two discrete time steps k and $k+1$, can be expanded as

$$\int_{(\Delta T)k}^{(\Delta T)(k+1)} \frac{1}{m_{\text{chaser}}}\Phi\left(\tau-(\Delta T)k\right)\begin{bmatrix}\mathbf{0}\\ \boldsymbol{I}\end{bmatrix}\boldsymbol{u}_v^L(\tau)d\tau = \int_{(\Delta T)k}^{(\Delta T)k+\overline{\Delta}} \frac{1}{m_{\text{chaser}}}\Phi\left(\tau-(\Delta T)k\right)\begin{bmatrix}\mathbf{0}\\ \boldsymbol{I}\end{bmatrix}\boldsymbol{u}_v^L(k)d\tau = \boldsymbol{B}\left(\overline{\Delta}\right)\boldsymbol{u}_v^L(k). \tag{23}$$

Here the matrix $\mathbf{B}\left(\overline{\Delta}\right)$ is obtained by integrating the Taylor expansion of the transition matrix $\Phi\left(\tau-(\Delta T)k\right)$ over the

time duration of the pulses $\overline{\Delta}$, in such way that

$$\boldsymbol{B}\left(\overline{\Delta}\right) = \frac{1}{m_{\text{chaser}}}\left(\overline{\Delta}\boldsymbol{I} + \frac{\overline{\Delta}^2\boldsymbol{A}}{2} + \frac{\overline{\Delta}^3\boldsymbol{A}^2}{6} + \ldots\right)\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{bmatrix}. \tag{24}$$

The matrix $\boldsymbol{A}$ above is the matrix representing the HCW dynamics, which is constant for a given reference orbit [25].

To propagate the full behavior of the system over the entire time horizon of the mission using algebraic operations we then collect all the states corresponding to the N_{step} time steps in this horizon within a vector $\boldsymbol{X}(k)$ with dimensions of $\left[6 \cdot N_{\text{step}}, 1\right]$, and all the control actions in a vector $\boldsymbol{U}_v(k)$ of $\left[3 \cdot N_{\text{step}}, 1\right]$. This means organizing this information as:

$$\boldsymbol{X}(\Delta T k) \coloneqq \begin{bmatrix} \boldsymbol{x}^L\big((\Delta T)(k+1)\big) \\ \vdots \\ \boldsymbol{x}^L\big((\Delta T)(k+j)\big) \\ \vdots \\ \boldsymbol{x}^L\big((\Delta T)(k+N_{\text{step}})\big)\end{bmatrix} \qquad \boldsymbol{U}_v(\Delta T k) \coloneqq \begin{bmatrix} \boldsymbol{u}_v^L\big((\Delta T)k\big) \\ \vdots \\ \boldsymbol{u}_v^L\big((\Delta T)(k+j)\big) \\ \vdots \\ \boldsymbol{u}_v^L\big((\Delta T)(k+N_{\text{step}}-1)\big)\end{bmatrix}. \tag{25}$$

We moreover define the auxiliary matrices:

$$\boldsymbol{F}(\Delta T) \coloneqq \begin{bmatrix} \Phi\left(\Delta T\right) \\ \vdots \\ \Phi\left(j\Delta T\right) \\ \vdots \\ \Phi\left(N_{\text{step}}\Delta T\right)\end{bmatrix} \tag{26}$$

$$\boldsymbol{H}\left(\Delta T, \overline{\Delta}\right) \coloneqq \begin{bmatrix} \boldsymbol{B}\left(\overline{\Delta}\right) & & & & \\ \Phi(\Delta T)\boldsymbol{B}\left(\overline{\Delta}\right) & \boldsymbol{B}\left(\overline{\Delta}\right) & & & \\ \Phi(2\Delta T)\boldsymbol{B}\left(\overline{\Delta}\right) & \Phi(\Delta T)\boldsymbol{B}\left(\overline{\Delta}\right) & \boldsymbol{B}\left(\overline{\Delta}\right) & & \\ \vdots & \vdots & \vdots & \ddots & \\ \Phi((N_{\text{step}}-1)\Delta T)\boldsymbol{B}\left(\overline{\Delta}\right) & \Phi((N_{\text{step}}-2)\Delta T)\boldsymbol{B}\left(\overline{\Delta}\right) & \ldots & \ldots & \boldsymbol{B}\left(\overline{\Delta}\right)\end{bmatrix} \tag{27}$$

so to be able to propagate the state of the chaser as:

$$\boldsymbol{X}(\Delta T k) = \boldsymbol{F}(\Delta T)\boldsymbol{x}^L(\Delta T k) + \boldsymbol{H}\left(\Delta T, \overline{\Delta}\right)\boldsymbol{U}_v(\Delta T k). \tag{28}$$

We note that $\boldsymbol{H}\left(\Delta T, \overline{\Delta}\right)$ is lower triangular, and that there exists an analytical expression for the state transition matrix $\Phi(t)$. These two facts lift from the need to use powered matrices as in [26]; this is beneficial both from computational perspectives and for avoiding numerical errors due to near-zero components.

Furthermore, these matrices depend only on the sampling period ΔT, the duration of the impulse $\overline{\Delta}$, and the length of the prediction horizon N_{step}. This implies that they can be computed once during the initialization of the MPC procedure, and then be stored in the chaser's memory before launch. This lifts the computation of the predictions from the need of using numerical integration schemes, leading to a sensible reduction of the computational cost of the MPC iterations.

To compact the notation, in the following we will let without loss of generality $\Delta T = 1$. Note that in the following we will moreover use t as a time index in continuous-time settings, k in discrete-time settings, and j to denote specific slots within the MPC horizon.

To guarantee the convexity of the MPC cost formulation, and in this way enable the use of convex optimization solvers to further speed up the computations, we then propose to consider as cost function

$$J(k) = \|\boldsymbol{U}_v(k)\|_{\boldsymbol{R}} + \sum_{j=1}^{N_{\text{step}}} \left\|\boldsymbol{x}^L(k+j) - \boldsymbol{x}^L_{\text{reference}}(j)\right\|_{\boldsymbol{Q}}. \tag{29}$$

Here $\boldsymbol{U}_v(k)$ is the vector of decision variables, $\boldsymbol{x}^L(k+j)$ is the chaser states vector subject to the dynamics defined by (28), $\boldsymbol{x}^L_{\text{reference}}(j)$ is a reference trajectory that will be constructed in the remainder of the paper, and $\|\cdot\|_{\boldsymbol{Q}}$ and $\|\cdot\|_{\boldsymbol{R}}$ are opportune weighted norms induced by the symmetric non-negative definite weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. Dimensionally speaking, such matrices are 6×6 and $\left(3 \cdot N_{\text{step}}\right) \times \left(3 \cdot N_{\text{step}}\right)$ respectively, while the horizon length is N_{step}. Note that this is actually a parameter that trades off computational complexity with the possibility of planning longer maneuvers. Its tuning is discussed in Section V.

Note also that cost (29) is implicitly parametrized by the estimated initial state $\boldsymbol{x}^L(k)$. Finally, $\boldsymbol{X}(k)$ (and thus $\boldsymbol{x}^L(k+j)$) is actually just a placeholder for (28). This quantity may thus be substituted directly with the right-hand side of (28), so that the optimization can be addressed in a sequential approach.

Summarizing, the purpose of (29) is to find the actuation that will make the chaser track "as well as possible" (in the $\boldsymbol{Q}$- and $\boldsymbol{R}$-sense) a reference path $\boldsymbol{x}^L_{\text{reference}}(1 : N_{\text{step}})$ assumed to be given. The generation of such a reference path will then be described in the dedicated following Section IV. That reference path will be built so to account for the tumbling motion of the target and the possibility of collisions. Embedding the collision avoidance constraints in the control problem is one of the major contributions of this paper.

IV. Generating a collision-free reference path

This section proposes how to build the reference path $\boldsymbol{x}^L_{\text{reference}}$ that will be used in (29) to compute the actual actuation signals in a receding horizon fashion. We start by noticing that the construction of such path $\boldsymbol{x}^L_{\text{reference}}$ shall:

1) satisfy the constraints:
    1) be based on computationally efficient schemes;
    2) be computable starting from just two measurements from the navigation system, i.e., the current position of the chaser in the LVLH frame $\boldsymbol{r}^L(k)$, and the current state of the docking point in the same frame $\boldsymbol{x}^L_{\text{dock}}(k)$;
2) pursue the objectives:
    1) let the chaser dock on a selected zone of the target, as in (14) or (15);
    2) let the docking happen with low relative velocity, as in (16);
    3) let the rendezvous happen independently of the initial conditions (attitude, position, velocity, etc.) of the chaser and the target;
    4) prevent chaser-target collisions in other parts than the selected docking zone.

The final objective of the maneuver is thus to reach the docking state $\boldsymbol{x}^L_{\text{dock}}$, defined as

$$\boldsymbol{x}^L_{\text{dock}} \coloneqq \begin{bmatrix} \boldsymbol{r}^L_{\text{dock}} \\ \boldsymbol{v}^L_{\text{dock}} \end{bmatrix} \tag{30}$$

where $\boldsymbol{r}^L_{\text{dock}}$ and $\boldsymbol{v}^L_{\text{dock}}$ are defined by (11) and (12), respectively. Note that we do not require $\boldsymbol{x}^L_{\text{reference}}(N_{\text{step}}) = \boldsymbol{x}^L_{\text{dock}}$, i.e., the final state of the trajectory used as a reference in the MPC problem (29) to be the final docking state. This is because the fixed-length horizon of the MPC scheme N_{step} (whose actual value shall be discussed in detail in the results section) may be shorter than the time necessary to successfully and safely dock starting from the current state $\boldsymbol{x}^L(k)$. Actually, the time that is required to do so is not known before the chaser starts its mission. In other words, we propose to build $\boldsymbol{x}^L_{\text{reference}}(1 : N_{\text{step}})$ so that $\boldsymbol{x}^L_{\text{reference}}(N_{\text{step}}) = \boldsymbol{x}^L_{\text{dock}}$ *only* when this state is within reach and reachable in a safe way.

Moreover, since chasing the docking point $\boldsymbol{x}^L_{\text{dock}}$ directly may lead to collisions and low efficient performance (as the chaser would be unnecessarily changing movement directions while approximating the target, we propose to build $\boldsymbol{x}^L_{\text{reference}}$ as the union of four distinct trajectories, depending on the phase of the mission:

1) **the tracking phase,** that happens when the chaser is so far away from the target that it is more meaningful to track its center of mass, rather than the actual docking point. This phase will thus aim at carrying the chaser near the target trading off fuel consumption against the time to complete the mission. During this phase the reference may thus just be $\mathbf{0}$, so to promote making $\boldsymbol{x}^L(k)$ go to zero,
2) **the approaching phase,** described in detail in Section IV.A, whose purpose is to transition the reference

trajectory from the "track the center of mass" phase to the "track the landing point" one. During this phase thus the reference $\boldsymbol{x}^L_{\text{reference}}(1:N_{\text{step}})$ should be a non-null ad-hoc trajectory $\boldsymbol{x}^L_{\text{appr}}(1:N_{\text{step}})$ that connects $\mathbf{0}$ (i.e., what is promoted during the tracking phase) to the next trajectory,

3) **the synchronization phase,** described in detail in Section IV.B, that happens when the chaser is sufficiently close to the target that it is better to synchronize the own orbit with the tumbling motion of the target. During this phase the generated reference $\boldsymbol{x}^L_{\text{reference}}(1:N_{\text{step}})$ aims thus at moving the chaser into a relative trajectory that follows the tumbling of the debris and thus promote achieving a final docking by means of an ad-hoc trajectory $\boldsymbol{x}^L_{\text{sync}}(1:N_{\text{step}})$,

4) **the docking phase,** described in detail in Section IV.C, that happens when the target follows the tumbling motion of the debris to the point that the final docking state $\boldsymbol{x}^L_{\text{dock}}$ may be reached safely. During this phase the generated reference $\boldsymbol{x}^L_{\text{reference}}(1:N_{\text{step}})$ will correspond to the ad-hoc trajectory $\boldsymbol{x}^L_{\text{end}}(1:N_{\text{step}})$, for which $\boldsymbol{x}^L_{\text{end}}(N_{\text{step}}) = \boldsymbol{x}^L_{\text{dock}}$.

To summarize, the proposed strategy is to start by tracking the center of mass of the target, then smoothly transition into a synchronous orbit of the tumbling motion of the docking point, and then finally dock. This means that given the currently measured distance between target and chaser $d(k)$, the reference $\boldsymbol{x}^L_{\text{reference}}(k)$ to be used within the MPC scheme will be

$$\boldsymbol{x}^L_{\text{reference}}(1:N_{\text{step}}) = \begin{cases} 0 & \text{if} \quad g_{\text{appr}} \leq d(k) \\ \boldsymbol{x}^L_{\text{appr}}(1:N_{\text{step}}) & \text{if} \quad g_{\text{sync}} \leq d(k) < g_{\text{appr}}, \\ \boldsymbol{x}^L_{\text{sync}}(1:N_{\text{step}}) & \text{if} \quad g_{\text{end}} \leq d(k) \leq g_{\text{sync}}, \\ \boldsymbol{x}^L_{\text{end}}(1:N_{\text{step}}) & \text{if} \quad d(k) < g_{\text{end}} \end{cases} \tag{31}$$

for opportune thresholds g_{appr}, g_{sync}, and g_{end} defined in the following sections IV.A, IV.B and IV.C.

A. The reference path generated while in the approaching phase

Once again the condition $g_{\text{appr}} \leq d(k) \implies \boldsymbol{x}^L_{\text{reference}}(1:N_{\text{step}}) = \mathbf{0}$ defined in (31) expresses the tracking phase, i.e., the situation where the chaser is sufficiently far from the target that it is actually not important to track the docking point, but rather the center of mass of the target. $\boldsymbol{x}^L_{\text{sync}}(1:N_{\text{step}})$ shall instead be a reference that synchronizes the orbit of the chaser with the tumbling motion of the docking point on the target. As said above, the transition from the tracking phase to the synchronization one should be smooth, so to avoid instantaneous changes in the reference trajectory, and thus avoid actuation spikes that might also lead to saturation effects, likely achievable with small-sat thrusters. $\boldsymbol{x}^L_{\text{appr}}(1:N_{\text{step}})$ serves thus this purpose of connecting such two phases.

Assuming then $\boldsymbol{x}^L_{\text{sync}}(1:N_{\text{step}})$ to be given, we propose to let $\boldsymbol{x}^L_{\text{appr}}$ be a sigmoid-like reference trajectory, i.e., a

smooth pondering between $\mathbf{0}$ and $\boldsymbol{x}^L_{\text{appr}}$ depending on the current distance from the target. In other words, we propose

$$\boldsymbol{x}^L_{\text{appr}}(j) = (1 - \beta(k))\, \boldsymbol{x}^L_{\text{sync}}(j); \tag{32}$$

with

$$\beta(k) = \frac{1}{1 + \exp\left(-\alpha|\boldsymbol{r}^L(k)| + \alpha\gamma\theta_{\text{debris length}}\right)} \tag{33}$$

with α and γ two tuning parameters, and $|\boldsymbol{r}^L(k)|$ the chaser-target relative distance at time k that shall thus be re-computed at each MPC iteration. A representation of sigmoid function $\beta(k)$ is shown in Figure 5 for different values of α and γ. We note that the main purpose of the sigmoid-like function in (32) is to generate a smooth change between the two phases, rather than shaping them. With definition (32), the closer the chaser gets to the target, the more the former is driven to synchronize its motion with the tumbling dynamics of the target.

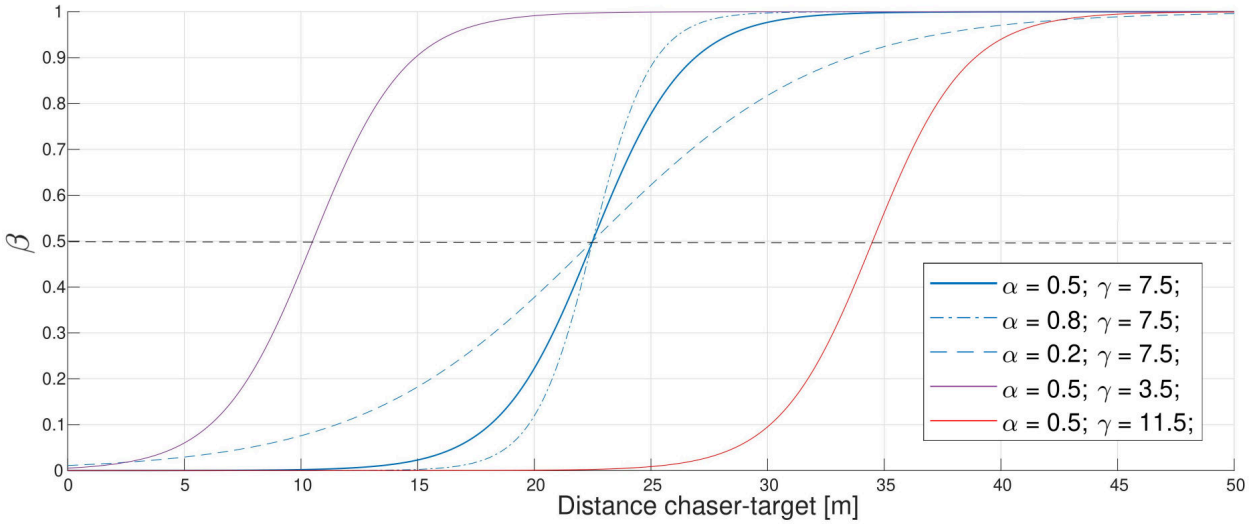


Fig. 5 Example of a sigmoids parameterized by α and γ.

We note that (32) may also be used to define formally the terms g_{appr} and g_{sync} as that boundaries in which $\beta(k)$ becomes one and zero within a certain tuning tolerance parameter ε, i.e.,

$$\text{given } \varepsilon,\ g_{\text{appr}} \text{ is such that } \frac{1}{1 + \exp\left(-\alpha g_{\text{appr}} + \alpha\gamma\theta_{\text{debris length}}\right)} = 1 - \varepsilon, \tag{34}$$

$$\text{given } \varepsilon,\ g_{\text{sync}} \text{ is such that } \frac{1}{1 + \exp\left(-\alpha g_{\text{sync}} + \alpha\gamma\theta_{\text{debris length}}\right)} = \varepsilon. \tag{35}$$

B. The reference path generated for the synchronization phase

As a starting note, it is meaningful to assume that during both the tracking and approaching phases g_{sync} in (31) is big enough so that there is no risk of chaser-target collisions. However, during the synchronization and final docking phases, this risk is present and needs to be accounted for. To embed collision avoidance while maintaining the convexity of the control problem, we propose to let the reference path $\boldsymbol{x}^L_{\text{reference}}(1 : N_{\text{step}})$ include such avoidance intrinsically through a constraints-relaxation-like approach, instead of via state constraints. Indeed the latter would naturally lead to non-convex and computationally intensive formulations, while the former has the advantage of reducing the complexity of more frequent low-level control computations.

To aid interpretability, the proposed $\boldsymbol{x}^L_{\text{reference}}$ will account for the concept of *collision sphere*, i.e., the sphere centered in the origin of the LVLH frame and of radius $\theta_{\text{debris length}}$ of all the possible points that may be reached by the docking point for an opportune random rotation of the target in the LVLH frame (graphically, the dark gray one in Figure 6). Such a collision sphere defines a "risk zone" that should not be entered during the synchronization phase (but only during the final docking one). To minimize this risk, we let the reference trajectory during the synchronization phase lay on a bigger co-centered sphere, caller *safety sphere*, and that corresponds graphically to the light gray one in Figure 6. Its radius is the one of the collision sphere multiplied by a safety factor $\theta_{\text{safety factor}} > 1$, so to give the maneuver some margin with respect to the collision sphere. Such a safety factor is another control parameter whose tuning will be discussed in Section V.

Considering then that we assume the target to be a cylindrical piece of debris, our proposal is then to let $\boldsymbol{x}^L_{\text{sync}}(1 : N_{\text{step}})$ be the geodesic along the safety sphere that starts at the projection of the current chaser position $\boldsymbol{r}^L(k)$ onto such safety sphere, and ends at the projection along the axis of the debris' cylinder of the docking point again onto the safety sphere (graphically, the dashed-dotted path in Figure 6). From intuitive perspectives, $\boldsymbol{x}^L_{\text{sync}}$ promotes synchronizing with the tumbling dynamics of the docking point while staying at a sufficiently far distance to avoid collisions and by ensuring that the initial states on the reference path never lay on the opposite side of the sphere. Note that then the trajectory never crosses the safety sphere.

For building such a path in a computationally efficient way we note that the first step is to compute the projection of the current position onto the safety sphere as

$$\boldsymbol{r}^L_{\text{safe}} \coloneqq \theta_{\text{safety factor}}\theta_{\text{debris length}}\frac{\boldsymbol{r}^L(k)}{|\boldsymbol{r}^L(k)|}. \tag{36}$$

To compute then $\boldsymbol{x}^L_{\text{sync}}(1 : N_{\text{step}})$ we use a Direction Cosine Matrix $\boldsymbol{T}(j)$ parameterized with the Euler vector and angle. This matrix is changing within the horizon (j), and uses as Euler vector $\boldsymbol{e}(j)$ the one normal to the plane formed by the current chaser state $\boldsymbol{r}^L(k)$ and the docking point for each propagated time $\boldsymbol{r}^L_{\text{dock}}(k + j)$. The turning angle $\alpha_\psi(j)$ is moreover distributed as a fraction $\Psi(j)$ of the angle between the docking point and chaser states and ends being the

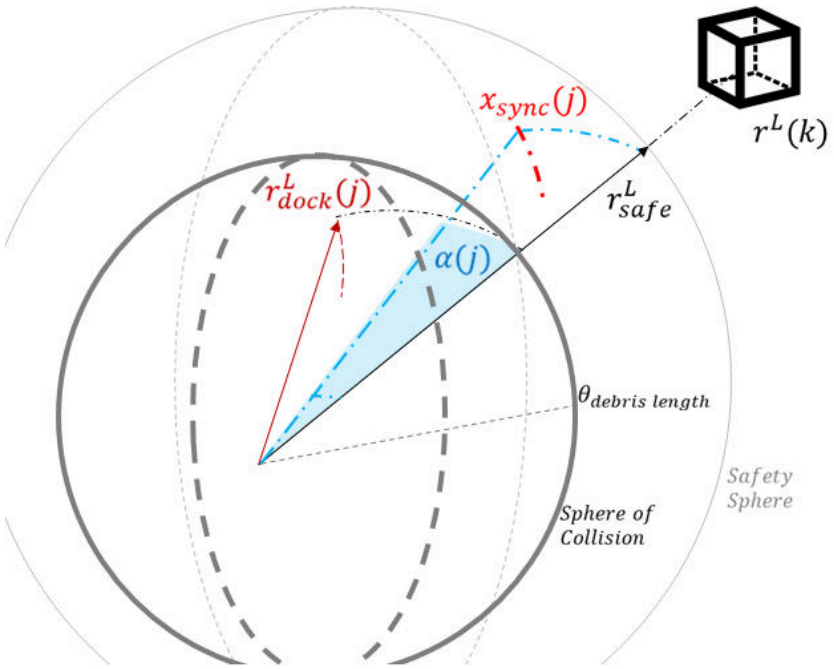


Fig. 6 A graphical intuition of the proposed reference trajectory during the synchronization phase.

total angle so that $\boldsymbol{x}^L_{\text{sync}}(N_{\text{step}})$ is the projection of the docking point at time $(k + N_{\text{step}})$ in the safety sphere. We propose function $\Psi(j)$ to have a square shape function of a control parameter ψ that determines the first fraction in the horizon.

$$\Psi(j) \coloneqq \sqrt{\psi + \frac{(j-1)}{(N_{\text{step}}-1)}(1-\psi)}; \tag{37}$$

$$\alpha_\psi(j) \coloneqq -\Psi(j)\cos^{-1}\left(\frac{\boldsymbol{r}^L(k)\cdot\boldsymbol{r}^L_{\text{dock}}(k+j)}{|\boldsymbol{r}^L(k)|\theta_{\text{debris length}}}\right); \quad c\alpha_\psi(j) \coloneqq \cos\alpha_\psi(j); \quad s\alpha_\psi(j) \coloneqq \sin\alpha_\psi(j); \tag{38}$$

$$\boldsymbol{e}(j) \coloneqq \frac{\boldsymbol{r}^L(k)\times\boldsymbol{r}^L_{\text{dock}}(k+j)}{\left|\boldsymbol{r}^L(k)\times\boldsymbol{r}^L_{\text{dock}}(k+j)\right|}; \tag{39}$$

$$\boldsymbol{T}(j) \coloneqq \begin{bmatrix} c\alpha_\psi(j)+\boldsymbol{e}(j,1)^2(1-c\alpha_\psi(j)) & \boldsymbol{e}(j,1)\boldsymbol{e}(j,2)(1-c\alpha_\psi(j))+\boldsymbol{e}(j,3)s\alpha_\psi(j) & \boldsymbol{e}(j,1)\boldsymbol{e}(j,3)(1-c\alpha_\psi(j))-\boldsymbol{e}(j,2)s\alpha_\psi(j) \\ \boldsymbol{e}(j,1)\boldsymbol{e}(j,2)(1-c\alpha_\psi(j))-\boldsymbol{e}(j,3)s\alpha_\psi(j) & c\alpha_\psi(j)+\boldsymbol{e}(j,2)^2(1-c\alpha_\psi(j)) & \boldsymbol{e}(j,2)\boldsymbol{e}(j,3)(1-c\alpha_\psi(j))+\boldsymbol{e}(j,1)s\alpha_\psi(j) \\ \boldsymbol{e}(j,1)\boldsymbol{e}(j,3)(1-c\alpha_\psi(j))+\boldsymbol{e}(j,2)s\alpha_\psi(j) & \boldsymbol{e}(j,2)\boldsymbol{e}(j,3)(1-c\alpha_\psi(j))-\boldsymbol{e}(j,1)s\alpha_\psi(j) & c\alpha_\psi(j)+\boldsymbol{e}(j,3)^2(1-c\alpha_\psi(j)) \end{bmatrix}, \tag{40}$$

thus, using (36) and (40), the synchronization reference path is given by

$$\boldsymbol{x}^L_{\text{sync}}(j) = \begin{bmatrix} \boldsymbol{T}(j)\boldsymbol{r}^L_{\text{safe}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{41}$$

We note that the algorithm above builds $\boldsymbol{x}^L_{\text{sync}}$ without using the velocity vector of the docking point over time. This means using less information, and from logical perspectives having made a choice relative to the trade-off "how much information needs to be measured vs. how much information is available for control purposes". Indeed, a control strategy that leverages $\boldsymbol{v}^L_{\text{dock}}$ would indeed potentially be more sophisticated, but at the same time require more weighting, tuning, and measuring efforts. At the same time the weight matrix $\boldsymbol{Q}$ ponders as zero the velocity components in $\boldsymbol{X}(k)$, and nevertheless, the trajectory is constructed by imposing a desired position each time step, that indirectly means chasing as well a velocity that converges to the one of the docking point in an approximate way, since the path is projected onto the safety sphere.

We also note that $\boldsymbol{x}^L_{\text{sync}}$ is implicitly parameterized by $\theta_{\text{safety factor}}$ and $\psi \geq 0$, that controls respectively the size of the safety sphere and the proximity between $\boldsymbol{x}^L_{\text{sync}}(1)$ and $\boldsymbol{r}^L_{\text{safe}}$. Using a value of $\psi \to 1$ may lead the chaser to cross the safety sphere and even the sphere of collision while pursuing $\boldsymbol{x}^L_{\text{sync}}$. On the other hand, a value of $\psi = 0$ decreases the performance of the strategy, leading to a very slow synchronization.

Ψ should not be confused with the actual angle difference between path and objective $\alpha_\psi(j)$. This angle gets smaller when the projections in the safety sphere are closer, which makes the path converge to the actual projection of the objective with time, as the chaser approaches the docking point and there is no risk of entering into the collision sphere by following the objective. Even so, the shape of the vector Ψ plays an important role in the performance. The square root shape selected has shown good behavior against linear distributions or sigmoid, but the controller can be modified to work with other functions.

C. The reference path generated while in the docking phase

The main objective of the reference trajectory in this phase is to simultaneously reduce the relative chaser-docking point distance $d(k)$ and velocity $v(k)$, and at the limit accomplish the rendezvous when such distances become zero.

Note the chaser may exit this phase even if it has previously entered it. Indeed disturbances and modeling errors may cause $d(k)$ to temporarily increase so that the condition $d(k) < g_{\text{end}}$ in (31) may be reversed during the final approach. We propose in this case to keep following (31), i.e., to change the reference trajectory again to the synchronization one, which will make the chaser gain distance from the target and prevent collisions with other parts of the target.

As for the value of g_{end} in (31), we propose

$$g_{\text{end}} = \theta_{\text{end phase}}\theta_{\text{debris length}} \quad \text{with} \quad \theta_{\text{end phase}} \geq (\theta_{\text{safety factor}} - 1).$$

The problem of tuning $\theta_{\text{end phase}}$ is postponed to Section V. Intuitively, this parameter shall be so that at a distance g_{end} a straight line joining the docking point and chaser position does not cross the sphere of collision. Therefore, g_{end} shall be in the order of the difference in ratio between safety and collision spheres. Figure 7 shows this distance that constitutes a spherical cone with a vertex in $\boldsymbol{r}^L_{\text{dock}}(k)$ and a side length of g_{end}.

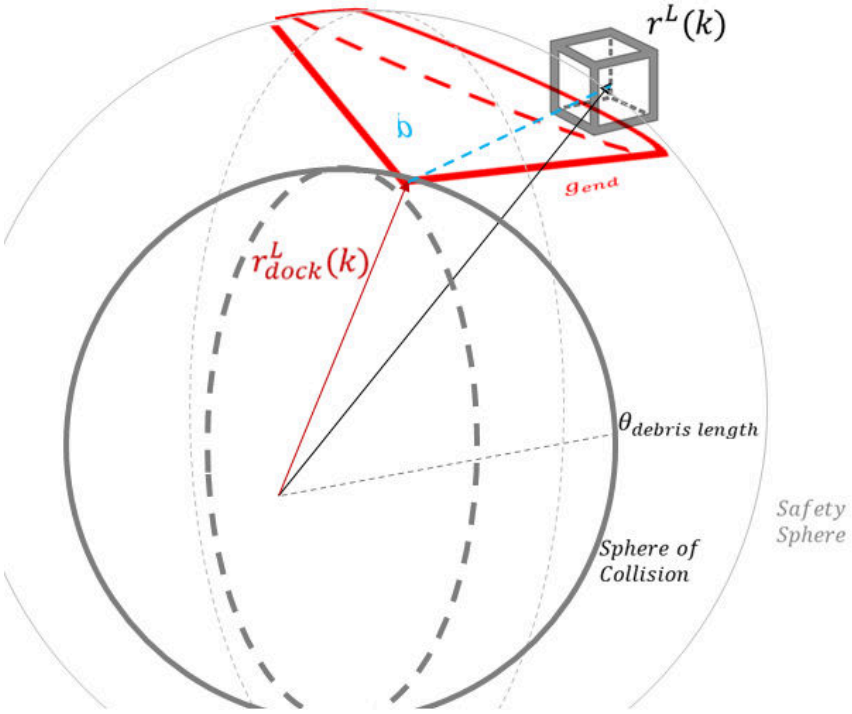


Fig. 7 A graphical intuition of the g_{end} spherical cone.

As for the actual reference trajectory $\boldsymbol{x}^L_{\text{end}}(1 : N_{\text{step}})$, we ladder on (41) in the sense that we build a path in the same way as for the synchronization one, but this time letting the sphere containing the path shrink in time until it becomes the collision sphere. This means that $\boldsymbol{x}^L_{\text{end}}(j)$ will follow an expression like (41), but where instead of considering a radius $\boldsymbol{r}^L_{\text{safe}}$ as in (36) the radius is now

$$\boldsymbol{r}^L_{\text{end}}(j) \coloneqq \theta_{\text{end}}(j)\frac{\boldsymbol{r}^L(k)}{|\boldsymbol{r}^L(k)|} \tag{42}$$

and the shrinking radius $\theta_{\text{end}}(j)$ decreasing smoothly as the sigmoid shaped function

$$\theta_{\text{end}}(j) = \Big((1-f(j))|\boldsymbol{r}^L(k)|+f(j)\theta_{\text{debris length}}\Big); \qquad f(j) = \frac{1}{1+\exp\left(-\alpha'|\boldsymbol{r}^L(k+j)| + 0.5\alpha'\theta_{\text{debris length}} + 0.5\alpha'|\boldsymbol{r}^L(k)|\right)} \tag{43}$$

so that, eventually,

$$\boldsymbol{x}^L_{\text{end}}(j) = \begin{bmatrix} \boldsymbol{T}(j)\boldsymbol{r}^L_{\text{end}}(j) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{44}$$

As for the parameters defining the strategy above, we shrink and modify conveniently the sigmoid in (43) be so that $\theta_{\text{end}}(1) \approx 0.99|\boldsymbol{r}^L(k)|$, imposing that the first element of the final reference trajectory is a slight reduction of the current position $\boldsymbol{r}^L(k)$, and $\theta_{\text{end}}(N_{\text{step}}) \approx \theta_{\text{debris length}}$ so that in combination with $\boldsymbol{T}(N_{\text{step}})$ the last element of this reference is the docking point. α' in (43) is a coefficient that measures the *spiral shape* of the docking reference trajectory and is such that the smaller it is, the slower the chaser descends into the docking point. α' thus trades off mission time and fuel consumption vs. smoothness in the relative velocity reduction. Starting the value of f for the first path state at 0.99 also means that a big horizon percentage of the path is indeed the motion followed by the docking point over the collision sphere. This is something beneficial for the way of the MPC works, as it means that the controller is also considering the task of maintaining this motion around the origin of the LVLH frame, so the control is computed to maintain the docking point position even if the actual docking is not performed at the predicted time.

V. Simulation setup and control parameters

In order to assess the performance of the proposed strategy, we have built a simulation tool in Matlab. Such a tool has the twofold function of reproducing a realistic scenario that includes the major effects affecting the relative dynamics of the rendezvous scenario, as described in Section II, and executing the MPC strategy, as described in Sections III and IV. In the following subsections, we illustrate some key features and settings that will be used to produce the numerical results.

A. Defining a specific rendezvous mission

To create realistic simulations, we select the target leveraging previous studies about the different dangers associated with different existent debris in space. To assess these dangers, we follow the strategy proposed in [2], which leads to the results summarized in table 1. Here the higher the ranking index R_N, the higher priority should be given to de-orbit that object.

Table 1 Risk indexes associated with spent upper stages currently in LEO, as ranked in [2].

Type of stage	Stages in LEO	Total mass in LEO [kg]	Total R_N	Mean R_N
Kosmos (SL-8) Second Stage	290	416150	239.12	0.82
Zenit-2 (SL-16) Second Stage	22	198000	810.75	36.85
Tsiklon-3 (SL-14) Third Stage	110	154770	37.59	0.34
Vostok (SL-3) Second Stage	53	58300	37.12	0.7
Delta 1 & 2 Second Stage	54	47375	11.47	0.21
Total	529 (64.4%)	874595 (72.9%)	1136.05	2.15

Note that *Zenit-2 second stages* score the highest R_N value of the Table 1, however, *Kosmos* stages are upper in the list due to their high number of such objects in LEO. This feature implying that one may repeat the same type of mission several times for such objects while learning and improving as one goes, made us select *Kosmos* as the ideal target for a debris removal mission. Furthermore, their reduced dimensions compared to *Zenit-2* implies that both collision avoidance and rendezvous are easier to perform. Indeed, for the same angular velocity larger objects have larger docking point velocities and thus higher risks of collision. This makes them an easier first target for demonstrating the here proposed rendezvous strategy.

We also note that [2] provides useful information such as orbital parameter distribution and geometric & mass properties of the considered targets. All the presented rocket bodies are allocated in similar orbits, and (except *Zenit-2 (SL-16)*) they have similar dimensions or are smaller than the selected target. This means that the here presented in-silico results are extensible to them as well. Summarizing, we select as the ideal target the *Kosmos (SL-8)*, whose properties are: dry mass 1435 kg, length 6 m, diameter 2.4 m, inclination 73.9 degrees, and main altitude 883 km. Therefore, we set the geometrical parameters of the target as $\theta_{\text{debris length}} = 3$ m and $\theta_{\text{debris radius}} = 1.2$ m.

As for the chaser, we consider a $m_{\text{chaser}} = 20$ kg 12U CubeSat whose initial relative orbit is contained in the XY plane of the LVLH frame. Without loss of generality, we consider the beginning of the maneuver to be at the cross of the orbit with the $\hat{x}_L$ axis of the LVLH frame, with $x < 0$. This means $\boldsymbol{x}_0^L = [-50, 0, 0, 0, 2 \cdot n \cdot 50, 0]^T$, being n the target orbital velocity (rad/s), and the initial condition given in [m] and [m/s]. As for the docking condition, we consider a docking point area defined as a circle of radius $\varepsilon_p = 50$ cm around the docking point, and a maximum tolerable chaser-target collision speed while docking of $\varepsilon_v = 20$ cm/s. We note that the proposed scheme has been able to complete the rendezvous mission with success for any initial condition in our simulations - a sign of the robustness of the overall approach.

B. Environment simulation

The experiments consider then a realistic simulation environment that accounts for perturbations, integrated numerically in the equations of motion in Section II. More precisely, since the dynamics are referred relative to the target body, the disturbances have been modeled as *differential perturbations*, and they include J_2 perturbations, solar pressure perturbations, and atmospheric drag, all using data from [27] and modeling the chaser according to the *free molecular flow* theory [28].

Note that we ignore third-body perturbation effects: since we treat the disturbances as differential ones, third-body perturbation effects are negligible due to the assumed proximity between the chaser and target. Note also that we assume the orbit of the target to be Keplerian and circular. This has been made since the mission duration is sufficiently short and can be relaxed without disrupting the structure of the proposed scheme, at the cost of a more complex control scheme.

C. Control parameters

The proposed strategy is characterized by many parameters that may be tuned so to adapt to the specific rendezvous mission. A summary of all the parameters used in the controller, with their final values given after the tuning, can be found in Table 2. Such a tuning allows for having successful maneuvers with the target rotating up to 5 deg/s and maximum levels of thrust between 0.1 and 0.5 N. Note that in the table $\boldsymbol{R}$ is a diagonal matrix with identical components, and $\boldsymbol{Q}$ is a matrix with only the three first components of the diagonal different from zero. The time step is set to $\Delta T = 2$ seconds for the controller, with a maximum thrust pulse duration of $\overline{\Delta} = 1$ second and no delays in the actuation. N_{step} moreover is set as constant throughout all the phases and tuned so that the horizon in the end phase is long enough to include the whole end of the docking operation. Finally, all the remaining parameters were found by following a trial and error approach seeking for maximizing the success rates for ever-increasing tumbling ratios while keeping the actuators far from saturation at the docking instant, avoiding collisions, and reducing the overall consumption. Note pushing the parameters to avoid thrusters saturation at the *beginning* of the mission is not a beneficial strategy.

Table 2 Parameters of control for the MPC and their tuned values.

Parameter	Short description	Value after tuning
α	Sizer of Approach sigmoid slope	0.5
γ	Sizer of Approach sigmoid half point	$3{\cdot}\theta_{\text{safety factor}}$
ψ	Initial angle driver for synchronization phase	0.4
$\theta_{\text{safety factor}}$	Sizer of safety sphere	2.5
$\theta_{\text{end phase}}$	Sizer of distance for end phase	$1.2{\cdot}(\theta_{\text{safety factor}} - 1)$
α'	Sizer of End sigmoid slope	50
Q	Weight matrix for state	$10 \cdot \omega_{yz0}$
R	Weight matrix for control	1000
N_{step}	Number of steps in horizon	200
ΔT	Time step [s]	2

D. Solver selection

We implement the proposed MPC scheme in `CVX` [29, 30] using the `MOSEK` solver with an academic license, the latter selected due to its low computational time and reliability. Figure 8 shows the average computational time for running an MPC step as a function of the horizon length for the test case showed in Section VI, tested on a Windows 10 Enterprise x64 bits machine running MATLAB R2022a-academic use embedded with an Intel(R) Core(TM) 5-8500 CPU @ 3.00 GHz 3.00 GHz; 32.0 GB (RAM). [†]

[†]Please note that these results are just indicative since the test is performed with `tic` and `toc` MATLAB functions in a personal computer, so not in an isolated environment where all computer capabilities are being used neither in space-graded hardware.

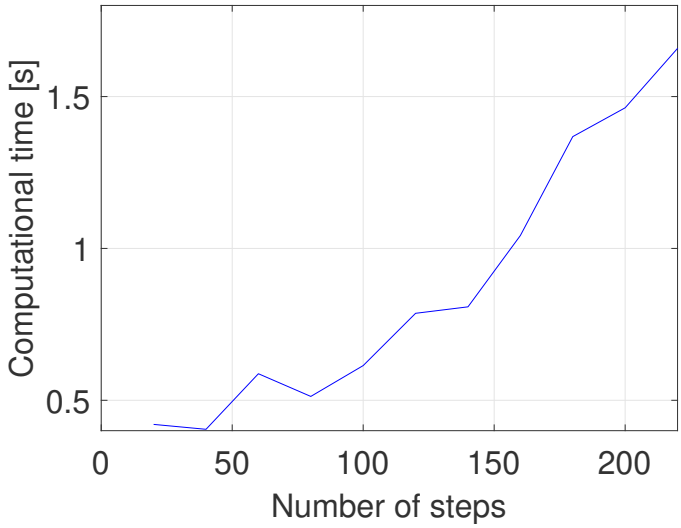


Fig. 8 Computational time of MPC as a function of the number of horizon steps.

VI. Results from simulations

To assess the effectiveness of our approach, we simulate a series of cases that range from the simplest ones without noise or perturbations to more complete ones including all the effects we expect to happen in a realistic environment. The simulations will graphically output results as in Figures 9 and 10. These represent snapshots in two different time instants of a rendezvous maneuver towards the rocket body. In such figures, the reference path $\boldsymbol{r}_{\text{reference}}$ corresponds to the dashed red line on the plots, and the corresponding actual trajectory followed by the chaser $\boldsymbol{r}^L$ up to that instant time is represented with the continuous red line, during the various phases of the rendezvous for a typical mission. In this specific case, Figures 9 and 10 show simulations for a mission for which the actuators of the chaser have their pulse thrust limited to 0.5 N, the target is rotating at $\omega_{yz0} = 3$deg/s and is spinning at $\omega_{x0} = 1$deg/s. The green line on the plots represents the motion followed by the docking point over the sphere of collision. This line intersects the reference path at the end of the maneuver, and after that intersection, the reference path becomes a propagation of this motion over the sphere of collision. In Figure 9, the represented reference trajectory corresponds to $\boldsymbol{r}_{\text{sync}}$, while in Figure 10 it represents $\boldsymbol{r}_{\text{end}}$.

In the following, we also plot results as in Figure 11, where one can visualize a complete example of a maneuver like in Figures 9 and 10 (in this specific case for the limit scenario with $U_{\max} = 500$ mN, $\omega_{yz0} = 3$ deg/s, perturbations, and noise in navigation inputs). The top plot of Figure 12 displays the actuation pulse duration $\Delta(k)$ using negative values for representing actuation in the negative direction of the axis. The second plot represents the chaser-docking point relative velocity v as in (13), while the third shows the relative distance between the chaser and target $D = |\boldsymbol{r}^L(k)|$. Note that saturation in the early beginning of the maneuvers is not due to the *approach sigmoid* tuning but rather to the

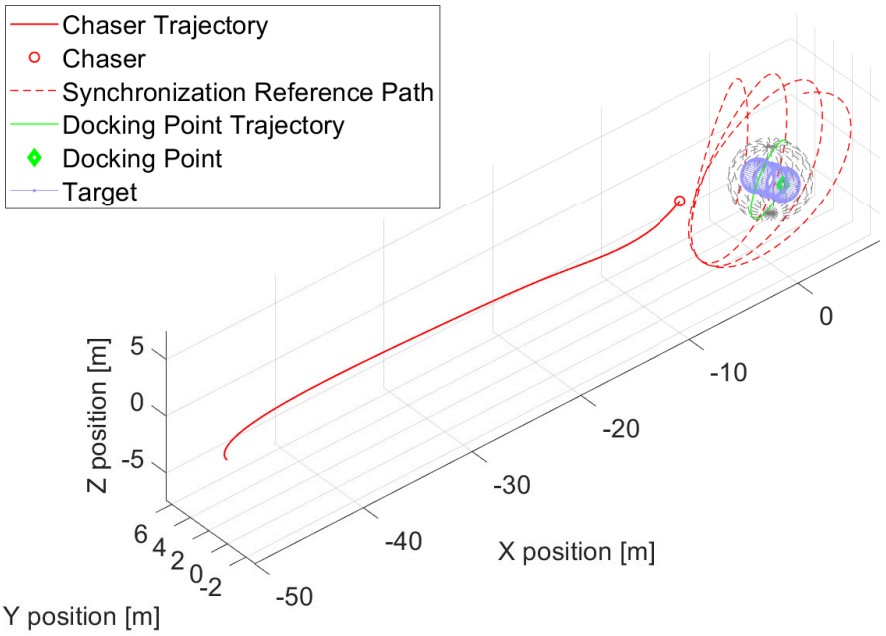


Fig. 9 Snapshot of the trajectory followed by the chaser during the synchronization phase.

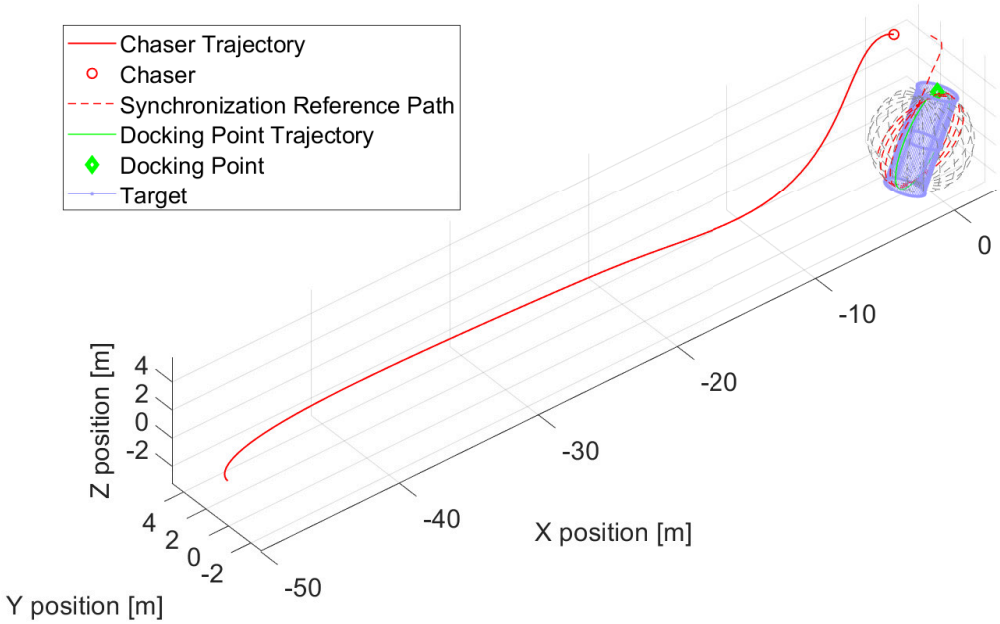


Fig. 10 Snapshot of the trajectory followed by the chaser during the near-docking instant.

optimization cost objective. Saturation at the beginning is not critical for collision avoidance and in any case, reduces the maneuver time.

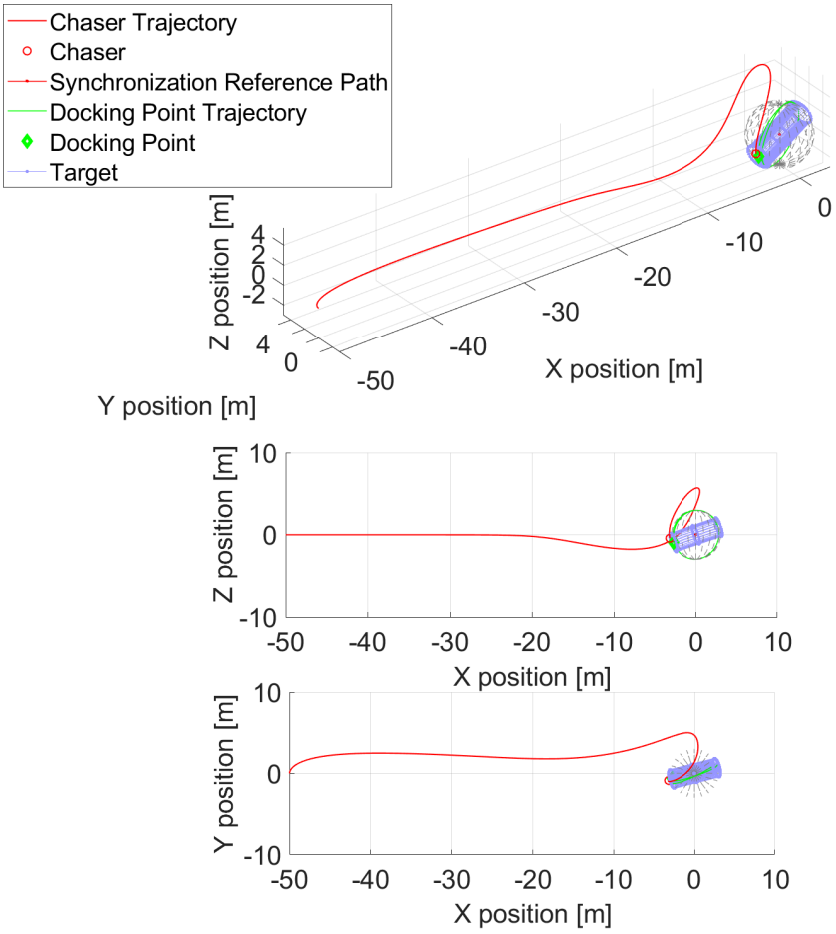


Fig. 11 Trajectory followed by the chaser for the case with $U_{\max} = 500$ mN, $\omega_{yz0} = 3$ deg/s, perturbations, and noise in navigation inputs.

To assess the robustness of the proposed MPC strategy against different tumbling conditions of the target we then perform the following Monte Carlo analysis: for each tumbling rate ω_{yz0} of the target and level of maximum thrust $U_{\max}$ of the chaser, we generate and simulate one hundred missions by randomly changing the initial attitude and tumbling direction of the target. The output of interest from this simulation are:

- the final docking distance from the center of the debris $D = \left|\boldsymbol{r}^L(t_f)\right|$, a quantity that gives an indication of the accuracy of the docking;
- the relative docking velocity $V = v(t_f)$, that indicates the level of impact that the chaser might have with the target;
- the total impulse I spent during the maneuver, which quantifies the eventual propellant consumption to perform the maneuver.

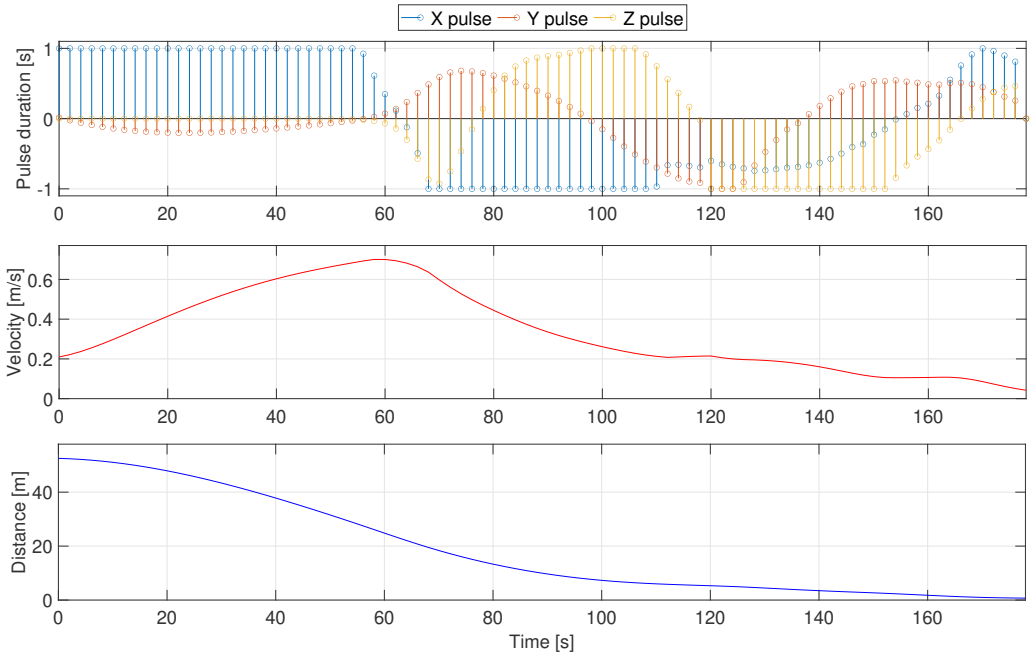


Fig. 12 Control action (expressed as pulse duration, relative velocity, and distance between the chaser and the target.

The results are then summarized in Tables 3 to 8 of Sections VI.A to VI.C, for which the columns headers μ_D, σ_D, μ_V, σ_V, μ_I, σ_I indicate mean and standard deviation for each of the outputs above, for different values of tumbling rates ω_{yz0} and maximum thrust $U_{\max}$.

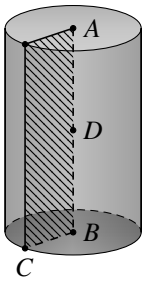
Figure 13 aids in interpreting the results and the evaluation of the success of the strategy for each combination. Here, μ_D is the average final docking distance from the center of the debris D and, thus, an indication of the expected accuracy of the docking maneuver for a specific mission configuration. We also introduce a new parameter L_c to measure the distance of the border of the cylinder from the center of the LVLH frame, represented as $\overline{DC}$ in Figure 13, which can be computed thus as

$$L_c = \sqrt{\theta^2_{\text{debris length}} + \theta^2_{\text{debris radius}}} \,. \tag{45}$$

We then postulate that for any given tumbling rate of the target there exist two different levels of success:

- if the mean docking distance μ_D obtained from the Monte Carlo simulations above is so that $\mu_D \in \left[\theta_{\text{debris length}}, L_c\right]$, then we say that the level of thrust of the chaser is enough to expect a successful accomplishment of the maneuver, i.e., with the chaser most likely ending on the external surface of the target for that specific debris scenario;
- if the mean docking distance is so that $\mu_D > L_c$, then we say that the chaser can dock very close to the docking point on the base of the target.

For the problem under consideration: $\theta_{\text{debris length}} = 3$ m, and $L_c = 3.2311$ m.



$\overline{AB} = 2\theta_{\text{debris length}}, \quad \overline{DC} = L_c$

Fig. 13 A sketch to aid in interpreting the Monte Carlo simulations about the docking results.

A. Monte Carlo results for the case of non-perturbed environments

Table 3 Statistics of the distribution of the various docking performance for non-perturbed dynamics scenarios.

$U_{\max}$ [N]	ω_{yz0} [deg/s]	μ_D [m]	σ_D [m]	μ_V [cm/s]	σ_V [cm/s]	μ_I [Ns]	σ_I [Ns]
0.1	1	3.2500	0.18544	3.4551	2.0671	24.1918	3.2991
0.1	2	3.1755	0.29098	9.0047	4.2064	25.6824	2.8099
0.1	3	1.8570	0.63208	14.5987	4.4178	25.4808	1.1908
0.5	1	3.3212	0.01684	2.1702	0.8440	37.3236	1.6971
0.5	2	3.3326	0.04059	3.0697	2.3185	42.3990	4.4466
0.5	3	3.2155	0.40765	6.4587	4.2461	54.6098	8.6187
0.5	4	2.9722	0.47482	8.1105	2.7193	57.3728	4.8152
0.5	5	2.9454	0.46851	17.9917	8.1561	44.2239	6.4120
0.5	6	2.1783	0.90189	20.0171	8.7252	44.4363	4.09831

We start reporting our results with the ideal case of non-perturbed environments, whose first and second-order statistics are summarized in Table 3, and for which we also plot examples of empirical distributions in Figure 14 (together with their Gaussian approximation). From Table 3 we can assess the performance of the controller in terms of maximum thrust-maximum tumbling rates:

- for a level of $U_{\max}$ = 100 mN, the maximum achievable tumbling target is rotating at 2 deg/s;
- for a level of $U_{\max}$ = 500 mN, the maximum achievable tumbling target (being conservative) is rotating at 3 deg/s;
- 4 deg/s tumbling targets are still possible candidates for $U_{\max}$ = 500 mN if docking on a target lateral surface is allowed;
- for higher tumbling rates the actuators get continuously saturated and the maneuver becomes unsuccessful.

Below the limits, the docking is produced in the zone highlighted in green in Figure 1.

1. Analysing the limit cases relative to these simulations

Figure 14 shows the presence of some outliers, i.e., simulations that fall somehow outside the typical simulated behavior. We note that the presence of remarkable outliers happens typically for the most critical missions, i.e., when the tumbling rate is close to the point for which the strategy cannot guarantee a successful docking, and especially when

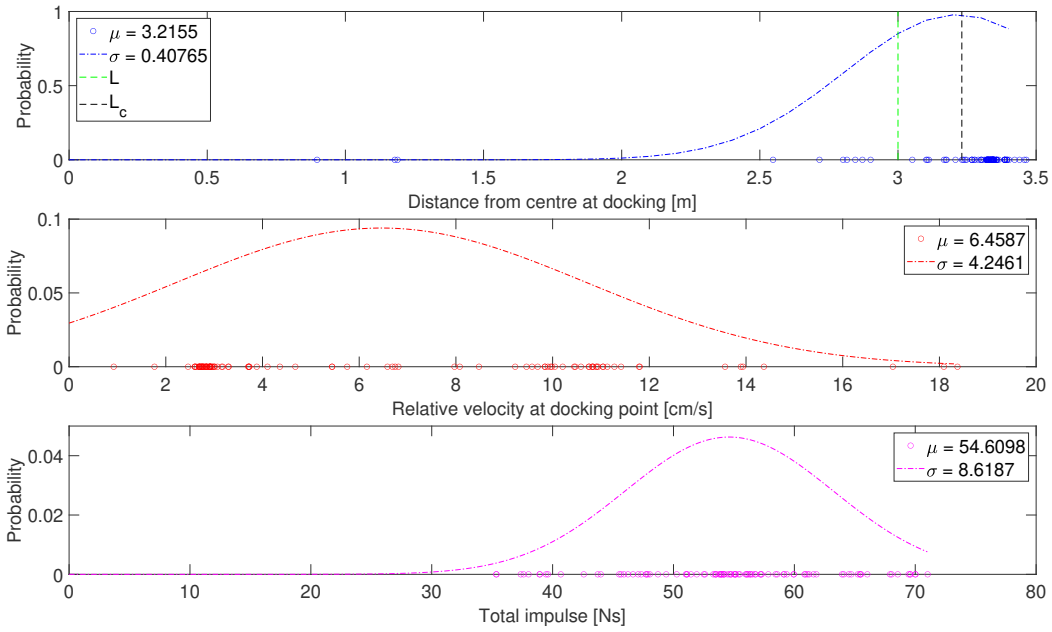


Fig. 14 Monte Carlo results obtained for the non-perturbed dynamics with $U_{\max} = 0.5$ N and $\omega_{yz0} = 3$ deg/s.

the debris is rotating in an opposite direction to the natural relative trajectories in space, plus when the initial attitude of the target is unfavorable.

More specifically, we define as natural motion a situation such that

$$\frac{v_y^L}{|v_y^L|} = -\frac{x^L}{|x^L|} \tag{46}$$

for relative orbits centered in the origin (an object goes faster when closer to the Earth). If the docking point is following a motion for which

$$\frac{v_y^L}{|v_y^L|} \approx \frac{x^L}{|x^L|}, \tag{47}$$

though, the controller has to fight against gravity while trying to change quickly its trajectory. This causes higher actuation demands, and increases the risk of saturation. The worst situation for a given tumbling rate is when this motion happens along the XY plane, for which this effect is more accentuated. On top of this, the initial attitude of the target plays a role too in worsening the situation for the chaser. This corresponds though only to a small range of degrees for which the initial attitude of the target is detrimental to the mission's success.

In these cases, it seems that selecting the time when to start the maneuver helps noticeably increase the chances of mission success. In other words, at the limit situations depending on the tumbling rate described above, for which the docking approach proposed in this paper sometimes is successful but sometimes not, depending on the initial conditions

at the start of the maneuver, it is beneficial to change the mission so that the chaser starts by getting close to the debris and following it from a distance while measuring its tumbling motion and eventually communicating the information to mission control. Mission control then carries out simulations and computes which favorable initial conditions should happen to start the mission, sending this information back to the chaser. Finally, the chaser locally estimates when conditions are suitable and start the mission accordingly. Note that such favorable initial conditions cannot be calculated analytically, since they rely on the tumbling rate, initial conditions, level of thrust, and rest of the control parameters; they thus need to be delegated to units with sufficiently powerful computing capabilities.

An example of results from this approach is shown in Figure 15, where we consider a scenario with the tumbling motion contained in the XY plane. Here triangles and dotted lines are for initial conditions in a stable point in $\hat{y}_L = -50$ m. It can be seen that there exists a specific angle leading to a bad mission (90°). However, with the proposed approach mission control may instruct the chaser to avoid such situations.

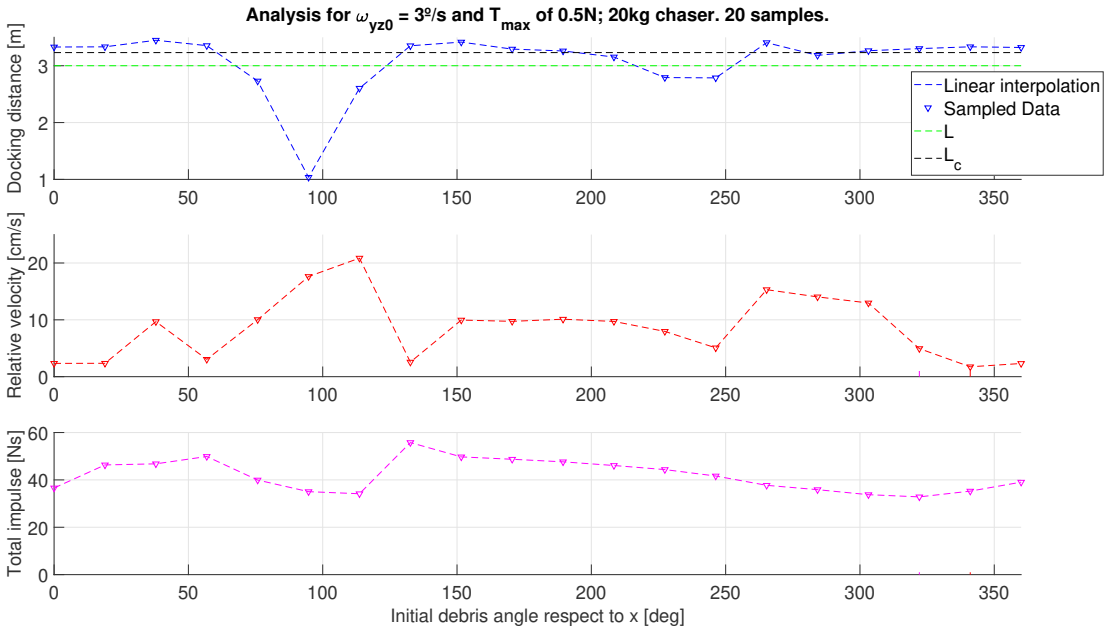


Fig. 15 Docking distance, relative velocity at docking, and Total Impulse with the initial relative attitude for $U_{\max} = 0.5$ N, $\omega_{yz0} = 3$ deg/s.

B. Monte Carlo results for the case of perturbed environments

We then analyze the effects of adding perturbations to the simulations above, and report the corresponding results in Table 4 again for the limit scenario of $U_{\max} = 0.5$ N and $\omega_{yz0} = 3$ deg/s. Note that all the effects are evaluated together in the same table, while a spinning rate is added in the last column to justify its low effect on the mission success rates.

Table 4 suggests that perturbations decrease the mission performance only slightly, thus not impeding docking both when taken individually and when considered together. More precisely, we note a general reduction in the docking

Table 4 Docking performance in perturbed scenarios with $U_{\max} = 0.5$ N and $\omega_{yz0} = 3$ deg/s.

Perturbation	μ_D [m]	σ_D [m]	μ_V [cm/s]	σ_V [cm/s]	μ_I [Ns]	σ_I [Ns]
None	3.2155	0.40765	6.4587	4.2641	54.6098	8.6187
$J_2(\theta_0 = 0^o)$	3.1523	0.48877	7.6656	4.7102	55.7396	7.5941
$J_2(\theta_0 = 90^o)$	3.1287	0.40819	7.1424	4.5373	51.7320	6.6491
Drag	3.1174	0.49234	8.5661	4.4502	56.8840	7.7615
Solar Pressure	3.1435	0.39631	7.4538	4.3732	52.7832	6.9872
All	3.0958	0.43944	7.5866	4.5156	51.4327	6.1110
All (+ $\omega_{x0} = 2$deg/s)	3.0770	0.46120	7.7929	4.6513	51.3860	5.9949

distance precision and increment of the relative velocity; at the same the typical standard deviations of the results and thus the overall distributions of the results seem unaffected. We also note that adding perturbations to the spinning motion of the debris does not add notable effects. Tables 5 to 7 show then the results for the rest of the feasible scenarios as reported in Table 3, again considering the presence of perturbations.

Table 5 Docking performance in perturbed scenarios with $U_{\max} = 0.5$ N and $\omega_{yz0} = 2$ deg/s.

Perturbation	μ_D [m]	σ_D [m]	μ_V [cm/s]	σ_V [cm/s]	μ_I [Ns]	σ_I [Ns]
None	3.3326	0.040585	3.0697	2.3185	42.3990	4.4466
All	3.3099	0.16687	2.7552	2.9318	47.3653	4.9343

Table 6 Docking performance in perturbed scenarios with $U_{\max} = 0.1$ N and $\omega_{yz0} = 2$ deg/s.

Perturbation	μ_D [m]	σ_D [m]	μ_V [cm/s]	σ_V [cm/s]	μ_I [Ns]	σ_I [Ns]
None	3.17550	0.29098	9.0047	4.2064	25.6824	2.8099
All	3.05495	0.46723	8.2198	4.12117	27.6593	2.7684

Table 7 Docking performance in perturbed scenarios with $U_{\max} = 0.1$ N and $\omega_{yz0} = 1$ deg/s.

Perturbation	μ_D [m]	σ_D [m]	μ_V [cm/s]	σ_V [cm/s]	μ_I [Ns]	σ_I [Ns]
None	3.2506	0.18544	3.4551	2.0671	24.1918	3.2991
All	3.3044	0.10414	2.7863	2.3929	26.4571	2.8845

From the tables above it seems thus that each differential perturbation on the chaser and debris has the same order of effects on the mission success and total impulse requirements, and thus no perturbation dominates the others. Moreover, perturbations seem to not affect too much the results we obtained in the unperturbed case. This means that not including such perturbations in the model considered by the proposed MPC was a reasonable choice, also because accounting for such perturbations in the model would imply having to measure such disturbances if one wants then to use that MPC in practice.

C. Analysing the effects of measurement noises

Measurement noise affects the control strategy by adding uncertainty to the estimate of the state of the system, an uncertainty that may then percolate on the mission success indicators. We simulate this percolation by adding normally distributed random noise on the main information used by the proposed docking scheme to build the reference trajectory $\boldsymbol{x}_{\text{reference}}$, i.e., the chaser position and velocity σ_{pc} and σ_{vc}, and the position of the docking point σ_{pt}. The results are then summarized in Table 8.

Table 8 Docking performance in perturbed scenarios for different noise levels with $U_{\max} = 0.5$ N and $\omega_{yz0} = 3$deg/s.

σ_{pt} [m]	σ_{pc} [m]	σ_{vc} [m/s]	μ_D [m]	σ_D [m]	μ_V [cm/s]	σ_V [cm/s]	μ_I [Ns]	σ_I [Ns]
0	0	0	3.2155	0.40765	6.4587	4.2641	54.6098	8.6187
0.01	0.01	0.001	3.0291	0.55510	8.9674	4.9140	56.2237	7.6262
0.01	0.1	0.001	3.0427	0.51351	9.6258	4.4336	54.8659	8.3827
0.01	1	0.001	2.4460	1.00570	17.1215	7.6008	119.1372	79.3582
0.1	0.01	0.001	3.0467	0.49356	9.1968	4.4457	57.3269	6.9313
1	0.01	0.001	2.9152	0.49127	7.6179	3.8108	49.5264	6.9005
0.01	0.01	0.01	2.857	0.60582	10.6684	5.0892	59.5415	8.442
0.1	0.1	0.001	2.9236	0.59103	9.9615	4.2063	55.7205	7.5316

Table 8 shows that the most critical noise is the one on the velocity of the chaser. Indeed we noted that sufficiently wrong measurements of such a quantity lead to a high risk of impacting the debris at high velocity. The overall recommendation that these results suggest is that good filtering of this information is essential for our proposed debris removal strategy.

VII. Conclusions

We proposed a debris docking approach that is based on combining an ad-hoc Model Predictive Control approach with a division of the mission in stages, and that embeds collision avoidance considerations. The overall strategy is computationally lightweight and thus can be run onboard by the chaser. Plus it is characterized by 10 control parameters, that enable tuning possibilities to fit specific tasks. We then showed in silico that tuning such controllers is not a difficult task and that the overall performance of the strategy is high for a variety of scenarios and random situations.

The simulations included uncertainties on the position and velocity of the chaser and of the target of up to 1 m and 1 cm/s respectively, distributed as Gaussian random noise to account for the fact that there is no perfect information available to the chasers at the beginning and during the mission. The levels of thrust that our algorithm requires, considering typical motions of known debris, are so that the approach can be followed using cold gas propulsion or similarly performing technologies. Electric Propulsion seems not suitable, instead.

Our simulations indicate moreover that collision avoidance with the debris while proceeding with the docking

mission is guaranteed until the rotation of the debris is ω_{yz0} = 3deg/s for the levels of $U_{\max}$ = 0.5 N, and ω_{yz0} = 1deg/s for $U_{\max}$ = 0.1 N. In other words, in all the seen setups for which the debris does not rotate more than these quantities, the chaser does not collide with the target while approaching it. This also means that assuming a chaser with a propulsion system capable of 500 mN, the proposed algorithm is so that the chaser cannot dock on debris whose rotations are higher than ω_{yz0} = 3deg/s. In these cases indeed the algorithm induces saturation in the actuation (something that impedes the mission). Moreover, if the tumbling rate is too high, there exist some unfavorable situations - and the higher the tumbling rate the more often these happen. These situations then benefit from or even require optimizing the starting time of the mission.

References

[1] Sagnières, L. B., Sharf, I., and Deleflie, F., "Simulation of long-term rotational dynamics of large space debris: A TOPEX/Poseidon case study," *Advances in Space Research*, Vol. 65, No. 4, 2020, pp. 1182–1195. https://doi.org/10.1016/j.asr.2019.11.021.

[2] Anselmo, L., and Pardini, C., "Ranking upper stages in low Earth orbit for active removal," *Acta Astronautica*, Vol. 122, 2016, pp. 19–27. https://doi.org/10.1016/j.actaastro.2016.01.019.

[3] Eren, U., Prach, A., Koçer, B. B., Raković, S. V., Kayacan, E., and Açıkmeşe, B., "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017, pp. 1541–1566. https://doi.org/10.2514/1.G002507.

[4] Leomanni, M., Rogers, E., and Gabriel, S. B., "Explicit model predictive control approach for low-thrust spacecraft proximity operations," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1780–1790. https://doi.org/10.2514/1.G000477.

[5] Leomanni, M., Bianchini, G., Garulli, A., Giannitrapani, A., and Quartullo, R., "Orbit Control Techniques for Space Debris Removal Missions Using Electric Propulsion," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 7, 2020, pp. 1259–1268. https://doi.org/10.2514/1.G004735.

[6] Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 4, 2015, pp. 1638–1647. https://doi.org/10.1109/TCST.2014.2379639.

[7] Malyuta, D., Reynolds, T., Szmuk, M., Acikmese, B., and Mesbahi, M., "Fast Trajectory Optimization via Successive Convexification for Spacecraft Rendezvous with Integer Constraints," *AIAA Scitech 2020 Forum*, 2020, p. 0616. https://doi.org/10.2514/6.2020-0616.

[8] Singh, L., Bortolami, S., and Page, L., "Optimal Guidance and Thruster Control in Orbital Approach and Rendezvous

for Docking using Model Predictive Control," *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7754. https://doi.org/10.2514/6.2010-7754.

[9] Arantes Gilz, P. R., Joldes, M., Louembet, C., and Camps, F., "Stable model predictive strategy for rendezvous hovering phases allowing for control saturation," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1658–1675. https://doi.org/10.2514/1.G003558.

[10] Leomanni, M., Quartullo, R., Bianchini, G., Garulli, A., and Giannitrapani, A., "Variable-Horizon Guidance for Autonomous Rendezvous and Docking to a Tumbling Target," *Journal of Guidance, Control, and Dynamics*, 2022, pp. 1–13. https://doi.org/10.2514/1.G006340.

[11] Larsén, A. K., Chen, Y., Bruschetta, M., Carli, R., Cenedese, A., Varagnolo, D., and Felicetti, L., "A computationally efficient model predictive control scheme for space debris rendezvous," *IFAC-PapersOnLine*, Vol. 52, No. 12, 2019, pp. 103–110. https://doi.org/10.1016/j.ifacol.2019.11.077, 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019.

[12] Sternberg, D., Hilton, A., Miller, D., McCarthy, B., Jewison, C., Roascio, D., James, J., and Saenz-Otero, A., "Reconfigurable ground and flight testing facility for robotic servicing, capture, and assembly," *2016 IEEE Aerospace Conference*, 2016, pp. 1–13. https://doi.org/10.1109/AERO.2016.7500526.

[13] Pong, C., Saenz-Otero, A., and Miller, D., "Autonomous thruster failure recovery on underactuated spacecraft using model predictive control," *Advances in the Astronautical Sciences*, Vol. 141, 2013. URL http://www.univelt.com/book=2771.

[14] Jewison, C., and Miller, D. W., "Probabilistic Trajectory Optimization Under Uncertain Path Constraints for Close Proximity Operations," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 9, 2018, pp. 1843–1858. https://doi.org/10.2514/1.G003152.

[15] Mammarella, M., Lorenzen, M., Capello, E., Park, H., Dabbene, F., Guglieri, G., Romano, M., and Allgöwer, F., "An Offline-Sampling SMPC Framework With Application to Autonomous Space Maneuvers," *IEEE Transactions on Control Systems Technology*, Vol. 28, No. 2, 2020, pp. 388–402. https://doi.org/10.1109/TCST.2018.2879938.

[16] Mammarella, M., Capello, E., Park, H., Guglieri, G., and Romano, M., "Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance," *Aerospace Science and Technology*, Vol. 77, 2018, pp. 585–594. https://doi.org/10.1016/j.ast.2018.04.009.

[17] Ortolano N., A. A., Geller D.K., "Autonomous Optimal Trajectory Planning for Orbital Rendezvous, Satellite Inspection, and Final Approach Based on Convex Optimization," *J Astronaut Sci*, Vol. 68, No. 2, 2021, pp. 444–479. https://doi.org/10.1007/s40295-021-00260-5.

[18] Richards A., J., How, "Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility," *IEEE American Control Conference*, Vol. 5, 2003, pp. 4034–4040. https://doi.org/10.1109/ACC.2003.1240467.

[19] Zhou, B.-Z., Liu, X.-F., and Cai, G.-P., "Motion-planning and pose-tracking based rendezvous and docking with a tumbling target," *Advances in Space Research*, Vol. 65, No. 4, 2020, pp. 1139–1157. https://doi.org/10.1016/j.asr.2019.11.013.

[20] Curtis, H. D., "Chapter 7 - Relative Motion and Rendezvous," *Orbital Mechanics for Engineering Students (Third Edition)*, edited by H. D. Curtis, Butterworth-Heinemann, Boston, 2014, third edition ed., pp. 367–404. https://doi.org/10.1016/B978-0-08-097747-8.00007-4.

[21] Hughes, P. C., *Spacecraft Attitude Dynamics, Chapter 4*, Dover Publications, 2012.

[22] Allan, D., Bates, C., Risbeck, M., and Rawlings, J., "On the inherent robustness of optimal and suboptimal nonlinear MPC," *System and Control Letters, Elsevier*, Vol. 106, 2017, pp. 68–78. https://doi.org/10.1016/j.sysconle.2017.03.005.

[23] NASA, "State-of-the-Art Small Spacecraft Technology," *NASA/TP—2020–5008734*, 2020, pp. 53–92.

[24] Vazquez, R., Gavilan, F., and Camacho, E. F., "Pulse-width predictive control for LTV systems with application to spacecraft rendezvous," *Control Engineering Practice*, Vol. 60, 2017, pp. 199–210. https://doi.org/10.1016/j.conengprac.2016.06.017.

[25] Alfriend, K. T., Vadali, S. R., Gurfil, P., How, J. P., and Breger, L. S., "Chapter 5 -Linear Equations of Relative Motion," *Spacecraft Formation Flying*, Butterworth-Heinemann, Oxford, 2010, pp. 84–90. https://doi.org/10.1016/B978-0-7506-8533-7.00219-0.

[26] de Bruijn, F. J., Theil, S., Choukroun, D., and Gill, E., "Geostationary Satellite Station-Keeping Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 605–616. https://doi.org/10.2514/1.G001302.

[27] Bruinsma, Sean L., Sánchez-Ortiz, Noelia, Olmedo, Estrella, and Guijarro, Nuria, "Evaluation of the DTM-2009 thermosphere model for benchmarking purposes," *J. Space Weather Space Clim.*, Vol. 2, 2012, p. A04. https://doi.org/10.1051/swsc/2012005.

[28] Storch, J., "Aerodynamic Disturbances on Spacecraft in Free-Molecular Flow," Tech. rep., Space and Missile Systems Center, Air Force Space Command, October 2002. https://doi.org/10.1061/40722(153)60.

[29] Grant, M., and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," http://cvxr.com/cvx, March 2014.

[30] Grant, M., and Boyd, S., "Graph implementations for nonsmooth convex programs," *Recent Advances in Learning and Control*, edited by V. Blondel, S. Boyd, and H. Kimura, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110. http://stanford.edu/~boyd/graph_dcp.html.



https://dspace.org/

Collision-avoiding model predictive rendezvous strategy to tumbling launcher stages

Ramírez, Jesús; Felicetti, Leonard; Varagnolo, Damiano

2023-08-01
American Institute of Aeronautics and Astronautics (AIAA)
https://doi.org/10.2514/1.G006804
Ramírez J, Felicetti L, Varagnolo D. (2023) Collision-avoiding model predictive rendezvous strategy to tumbling launcher stages. Journal of Guidance, Control, and Dynamics, Volume 46, Issue 8, August 2023, pp. 1564-1579

Downloaded from a DSpace repository. https://dspace.org/